

# **FEE TRANSMITTAL** **for FY 2005**

Effective 10/01/2004. Patent fees are subject to annual revision.

☐ Applicant claims small entity status. See 37 CFR 1.27

**TOTAL AMOUNT OF PAYMENT** (\$) **130.00**
**Complete if Known**

Application Number	10/663,480
Filing Date	September 15, 2003
First Named Inventor	Honda, Kiyoshi
Examiner Name	Unassigned
Art Unit	2186
Attorney Docket No.	16869P-078800US

**METHOD OF PAYMENT (check all that apply)**
☐ Check ☐ Credit Card ☐ Money Order ☐ Other ☐ None

☒ Deposit Account:

 Deposit  
Account  
Number

20-1430

 Deposit  
Account  
Name

Townsend and Townsend and Crew LLP

The Director is authorized to: (check all that apply)

☒ Charge fee(s) indicated below ☒ Credit any overpayments

☒ Charge any additional fee(s) or any underpayment of fee(s)

☐ Charge fee(s) indicated below, except for the filing fee to the above-identified deposit account.

**FEE CALCULATION**
**1. BASIC FILING FEE**

Large Entity		Small Entity		Fee Description	Fee Paid
Fee Code	Fee (\$)	Fee Code	Fee (\$)		
1001	790	2001	395	Utility filing fee	
1002	350	2002	175	Design filing fee	
1003	550	2003	275	Plant filing fee	
1004	790	2004	395	Reissue filing fee	
1005	160	2005	80	Provisional filing fee	

SUBTOTAL (1)

(\$0.00)

**2. EXTRA CLAIM FEES FOR UTILITY AND REISSUE**

Total Claims		Extra Claims		Fee from below		Fee Paid
Independent Claims		** =		X		
Multiple Dependent				X		

Large Entity		Small Entity		Fee Description
Fee Code	Fee (\$)	Fee Code	Fee (\$)	
1202	18	2202	9	Claims in excess of 20
1201	88	2201	44	Independent claims in excess of 3
1203	300	2203	150	Multiple dependent claim, if not paid
1204	88	2204	44	** Reissue independent claims over original patent
1205	18	2205	9	** Reissue claims in excess of 20 and over original patent

SUBTOTAL (2)

(\$0.00)

\*\*or number previously paid, if greater; For Reissues, see above

**FEE CALCULATION (continued)**
**3. ADDITIONAL FEES**

Large Entity		Small Entity		Fee Description	Fee Paid
Fee Code	Fee (\$)	Fee Code	Fee (\$)		
1051	130	2051	65	Surcharge - late filing fee or oath	
1052	50	2052	25	Surcharge - late provisional filing fee or cover sheet.	
1053	130	1053	130	Non-English specification	
1812	2,520	1812	2,520	For filing a request for <i>ex parte</i> reexamination	
1804	920*	1804	920*	Requesting publication of SIR prior to Examiner action	
1805	1,840*	1805	1,840*	Requesting publication of SIR after Examiner action	
1251	110	2251	55	Extension for reply within first month	
1252	430	2252	215	Extension for reply within second month	
1253	980	2253	490	Extension for reply within third month	
1254	1,530	2254	765	Extension for reply within fourth month	
1255	2,080	2255	1,040	Extension for reply within fifth month	
1401	340	2401	170	Notice of Appeal	
1402	340	2402	170	Filing a brief in support of an appeal	
1403	300	2403	150	Request for oral hearing	
1451	1,510	1451	1,510	Petition to institute a public use proceeding	
1452	110	2452	55	Petition to revive - unavoidable	
1453	1,330	2453	665	Petition to revive - unintentional	
1501	1,370	2501	685	Utility issue fee (or reissue)	
1502	490	2502	245	Design issue fee	
1503	660	2503	330	Plant issue fee	
1460	130	1460	130	Petitions to the Commissioner	130
1807	50	1807	50	Processing fee under 37 CFR 1.17(q)	
1806	180	1806	180	Submission of Information Disclosure Stmt	
8021	40	8021	40	Recording each patent assignment per property (times number of properties)	
1809	790	2809	395	Filing a submission after final rejection (37 CFR § 1.129(a))	
1810	790	2810	395	For each additional invention to be examined (37 CFR § 1.129(b))	
1801	790	2801	395	Request for Continued Examination (RCE)	
1802	900	1802	900	Request for expedited examination of a design application	

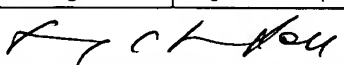
Other fee (specify)

\*Reduced by Basic Filing Fee Paid

SUBTOTAL (3)

(\$130.00)

**SUBMITTED BY**
**Complete (if applicable)**

Name (Print/Type)	Chun-Pok Leung	Registration No. (Attorney/Agent)	41,405	Telephone	650-326-2400
Signature				Date	November 17, 2004

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

In re application of:

KIYOSHI HONDA et al.

Application No.: 10/663,480

Filed: September 15, 2003

For: VIRTUALIZATION  
CONTROLLER AND DATA  
TRANSFER CONTROL  
METHOD

Customer No.: 20350

Examiner: Unassigned

Technology Center/Art Unit: 2186

Confirmation No.: 2877

**PETITION TO MAKE SPECIAL FOR  
NEW APPLICATION UNDER M.P.E.P.  
§ 708.02, VIII & 37 C.F.R. § 1.102(d)**

Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

Sir:

This is a petition to make special the above-identified application under MPEP § 708.02, VIII & 37 C.F.R. § 1.102(d). The application has not received any examination by an Examiner.

(a) The Commissioner is authorized to charge the petition fee of \$130 under 37 C.F.R. § 1.17(i) and any other fees associated with this paper to Deposit Account 20-1430.

(b) All the claims are believed to be directed to a single invention. If the Office determines that all the claims presented are not obviously directed to a single invention, then Applicants will make an election without traverse as a prerequisite to the grant of special status.

(c) Pre-examination searches were made of U.S. issued patents, including a classification search and a key word search. The classification search was conducted on or around October 7, 2004 covering Class 711 (subclasses 6, 111, 114, 154, 165, 170, 173, 202, and 203) and Class 718 (subclass 1), by a professional search firm, Lacasse & Associates, LLC. The key word search was performed on the USPTO full-text database including published U.S. patent applications. The inventors further provided five references considered most closely related to the subject matter of the present application (see references #6-10 below), which were cited in the Information Disclosure Statement filed with the application on September 15, 2003.

(d) The following references, copies of which are attached herewith, are deemed most closely related to the subject matter encompassed by the claims:

- (1) U.S. Patent No. 6,647,387 B1;
- (2) U.S. Patent No. 6,718,404 B2;
- (3) U.S. Patent Publication No. 2003/0204700 A1;
- (4) U.S. Patent Publication No. 2004/0054866 A1;
- (5) U.S. Patent Publication No. 2004/0068637 A1;
- (6) U.S. Patent No. 5,680,640;
- (7) U.S. Patent Publication No. 2001/0054133 A1;
- (8) Japanese Patent Publication No. JP 2001-249853;
- (9) Japanese Patent Publication No. JP 2001-331355 (U.S. 09/991,219); and
- (10) European Patent Publication No. EP 1130514 A2.

(e) Set forth below is a detailed discussion of references which points out with particularity how the claimed subject matter is distinguishable over the references.

A. Claimed Embodiments of the Present Invention

The claimed embodiments relate to transferring data between a plurality of storage devices without a host computer issuing an access request to a storage device being aware of the data transfer process.

Independent claim 1 recites a virtualization controller for controlling data transfer between a host system and a plurality of storage devices. The virtualization controller comprises a plurality of first ports for connection with the plurality of storage devices each having a storage area to store data; a second port for connection with the host system; a processor; and a memory configured to store volume mapping information which correlates first identification information used by the host system to access a first storage area in one of the storage devices, with second identification information for identifying the first storage area, the correlation being used by the processor to access the first storage area. When data stored in the first storage area is transferred to a second storage area in one of the storage devices, the processor correlates the first identification information with a third identification information for identifying the second storage area and registers the first identification information and the third identification information in the volume mapping information.

Independent claim 10 recites a data control system connected to one or more host systems. The data control system comprises a plurality of storage devices each having a storage area; and a switch which is connected with the plurality of storage devices and the one or more host systems. The switch includes a plurality of first ports for connection with the storage devices; one or more second ports for connection with the one or more host systems; a memory configured to store information on a correlation between first identification information used by the host system to access a first storage area of one of the storage devices, and second identification information for identifying the first storage area; and a routing processor configured to convert data with the first identification information received from the host system into data with the second identification information and to send the converted data to the storage device having the first storage area according to the



correlation information. When data stored in the first storage area is transferred to a second storage area of one of the storage devices, the routing processor converts data with the first identification information into data with third identification information for identifying the second storage area and sends the converted data to the storage device having the second storage area.

Independent claim 16 recites a method of controlling data transfer in a system including a host system which uses first identification information to access a first storage area in one of a plurality of storage devices, wherein the first storage area includes data associated with second identification information identifying the first storage area. The method comprises issuing a data transfer request to the first storage device to transfer the data with the second identification information in the first storage device to a second storage device; and upon receipt of notification of completion of data transfer from the first storage device to the second storage device, correlating the first identification information with a third identification information identifying the second storage area containing the transferred data.

Independent claim 19 recites a method of connecting a virtualization controller between a host system and a storage device which are connected through a first path between a first port of the host system and a first port of the storage device and a second path between a second port of the host system and a second port of the storage device. The method comprises accessing a storage area of the storage device; disconnecting the second path between the second port of the host system and the second port of the storage device; connecting the second port of the host system with the virtualization controller through a third path; connecting the virtualization controller with the second port of the storage device through a fourth path; and setting, on the virtualization controller, identification used by the host system to identify the storage area, identification information for the second port of the storage device, and virtual port identification information for the virtualization controller, which are correlated to define access of the storage area by the host system.

Independent claim 24 recites a method of controlling data transfer in a system including a host system which uses first identification information to access a first storage area in one of a plurality of storage devices, wherein the first storage area includes data associated with second identification information identifying the first storage area. The

method comprises receiving a first request with the first identification information from the host system to access the data associated with the second identification information identifying the first storage area; sending a second request with the second identification information to the first storage area; receiving data corresponding to the second request from the first storage area; and sending the received data to the host system.

One of the benefits that may be derived is that data can be transferred between storage devices without revising the identification information which a host computer uses to identify the volume to be accessed. As a consequence, data transfer takes place without stopping operation of the host computer. Furthermore, even when a virtualization controller is newly introduced or replaced in a computer system, a host computer can access data without revising the information for identifying the volume to be accessed. As a consequence, it is possible to introduce or replace a virtualization controller in the computer system without stopping operation of the host computer.

B. Discussion of the References

None of the following references disclose that when data stored in the first storage area is transferred to a second storage area in one of the storage devices, the processor correlates the first identification information with a third identification information for identifying the second storage area and registers the first identification information and the third identification information in the volume mapping information.

1. U.S. Patent No. 6,647,387 B1

This reference discloses a controller 114 which may be connected across local I/O interface 134 to memory 138 and port 136, controller 114 also appears to contain a processor 132. Processor 132 may fetch and execute computer program instructions and data from memory 138. Such computer program instructions and data may include, controller SAN management procedure 140 and controller port ID/LU mapping table 142. A method of correlating storage identifiers upon data transfer does not appear to be discussed. See Fig. 4; column 5 lines 32-33, lines 66-67; and column 6 lines 1-3.

2. U.S. Patent No. 6,718,404 B2

This reference discloses a controller 120, which may store mapping table 200 in a semi-permanent memory. Mapping table 200 may contain a mapping that relates position in a virtual disk with an actual location on storage devices. Data migration may be done in response to an automated policy decision to move virtual disk data from one physical storage location 230 to another. Controller 120 may copy contents of existing physical storage location 230 to a new physical location 230. The controller 120 may then update its persistently stored mapping table 200 to reflect the new storage location 230. Controller 120 does not appear to have direct port connections to the hosts or the storage devices. See Fig. 1; column 2 lines 3-5, lines 64-67; column 7 lines 17-19; column 7, line 61 to column 8, line 4.

3. U.S. Patent Publication No. 2003/0204700 A1

This reference relates to a controller 6 which may maintain a virtual storage map that describes current allocations of primary storage volume 12 and secondary virtual storage 14 within storage 8. Controller 6 may store the VPD map and VSM information within both memory 22 and storage system 8 for purposes of redundancy. In response to save/backup request, controller 6 updates VSM to include data written to secondary volume. See paragraph [0053] and Fig. 3.

4. U.S. Patent Publication No. 2004/0054866 A1

This reference discloses a storage controller 27, which may include a plurality of port adapters 35, 36. Port adapter 36 may include a microprocessor 235 and random access memory 236. Random access memory 236 may be programmed with volume access and mapping information 246. The volume access and mapping information may include a virtual port host table 281 listing each host having access rights through a virtual switch controlled with the volume access and mapping information, and a virtual port mapping table 282 listing each virtual port accessible through the virtual switch controlled with the volume access and mapping information. A method of correlating storage identifiers upon data transfer does not appear to be discussed. See Figs. 21 and 23; and paragraphs [0059], [0126], [0128], and [0139].

5. U.S. Patent Publication No. 2004/0068637 A1

This reference discloses a controller 12 which may include a central processing unit (CPU) 16 and a memory 18. Memory 18 may be utilized to store maps, for use in addressing storage space 14 and store executable code usable controller 12. Controller 12 may initiate a data migration request, and may modify pointer 34 associated with data migration request, using new physical address 36, so now pointer 34 addresses the migrated data at modified address 36. See Fig. 1; and paragraphs [0020], [0022], [0059], and [0060].

6. U.S. Patent No. 5,680,640

This reference discloses a system and method for providing on-line, real-time, transparent data migration from a first data storage system to a second data storage system. The first data storage system which had previously been coupled to a host, network or other data processing system is disconnected from the host and connected to a second data storage system. The second data storage system is coupled to the host or data processing system. The second data processing system includes a data map or table which indicates which data elements are stored on the second data storage system and which corresponding data elements on the first data processing device have been copied to the second data storage system. When the host, network or data processing system requests data from a data storage system, the second data storage system determines whether or not the data is stored on the second or first data storage system. If the data is stored on the second data storage system, the data is made available to the requesting device. If the data is not stored on a second data storage system, the second data storage system issues a request to the first data storage system, retrieves the data, makes the data available to the requesting device, writes the data to the second data storage system and updates the data element map or table. When not busy servicing other requests, the second data storage system scans the data map or table to determine which data elements from the first data storage device have not been copied to the second data storage device, and performs copying of the data and updating of the data map or table in the background, independent of any coupled host, network or other similar data processing device.

7. U.S. Patent Publication No. 2001/0054133 A1

This reference discloses a technique for efficient data transfer concerning a system in which a volume used by an application is provided over a plurality of storage areas which are controlled by different controllers. According to this technique, data in a volume which is used by a specific application is transferred preferentially. Data transfer in a storage device subsystem is executed by a controller which controls the storage device subsystem, and the controller can hide the data transfer process from a host computer connected with it. However, this method does not take into consideration data transfer which takes place over more than one storage device subsystem. Therefore, if data stored in a certain storage device subsystem is to be transferred to another storage device subsystem, the controller must inform the host computer that, in data transfer, the storage device subsystem to be accessed will change, which means that it is impossible to hide the data transfer process from the host computer. Also, this reference does not disclose any technique of hiding such data transfer process from the host computer. Consequently, for data transfer, the host computer must stop its operation once and specify the storage device or volume to be accessed again.

8. Japanese Patent Publication No. JP 2001-249853

This reference discloses a data migration method suitable for SAN environment and its devices. A disk device at migration destination is connected with a switch in which a host is connected with a disk device at a migration origin. At this memory, the same value as that of a port ID provided to an F-Port of the switch is provided to a port ID of the disk device at the migration destination and the disk device at the migration destination is made not to be recognized by the host 101. The disk information at the migration destination reads the constitutional information on the disk device at the migration origin. After that, a physical port ID 602 of the disk device 103 at the migration origin is replaced with the physical port ID 602 of the disk device 104 at the migration destination in the switch. The disk device at the migration destination constructs volume according to a logical volume number on the disk device at the migration origin and the size of the volume and stores data in the disk device at the migration origin in corresponding volume.

9. Japanese Patent Publication No. JP 2001-331355 (U.S. 09/991,219)

This reference discloses a system that is transparent to the hosts and the users, and that is capable of making full utilization of the scalability of SAN. The computer system includes host computers, a back end computer (back end server), a plurality of storage subsystems, and a switch for connecting at least the host computers with the back end computer. The host computers access each storage subsystem via the back end computer. The back end computer provides one virtual disk apparatus or a plurality of virtual disk apparatuses to the host computers. If the host computers issue access requests to the virtual disk apparatus/apparatuses, the back end computer issues an appropriate request to the storage subsystems connected thereto actually, depending on the type of the virtual disk apparatus/apparatuses to which the requests have been issued. This makes it possible to implement all the data manipulations, such as the data migration among the disk apparatuses and on-line extension of the disk capacities, completely transparently to the host computers.

10. European Patent Publication No. EP 1130514 A2

This reference contains the same disclosure as reference #8, Japanese Patent Publication No. JP 2001-249853.

(f) In view of this petition, the Examiner is respectfully requested to issue a first Office Action at an early date.

Respectfully submitted,



Chun-Pok Leung  
Reg. No. 41,405

TOWNSEND and TOWNSEND and CREW LLP  
Two Embarcadero Center, 8<sup>th</sup> Floor  
San Francisco, California 94111-3834  
Tel: 650-326-2400  
Fax: 415-576-0300  
Attachments  
RL:rl  
60351362 v1

## PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2001-249853

(43)Date of publication of application : 14.09.2001

(51)Int.Cl.

G06F 12/16

G06F 3/06

G06F 12/00

G06F 13/14

G11B 20/10

(21)Application number : 2000-063289

(71)Applicant : HITACHI LTD

(22)Date of filing : 03.03.2000

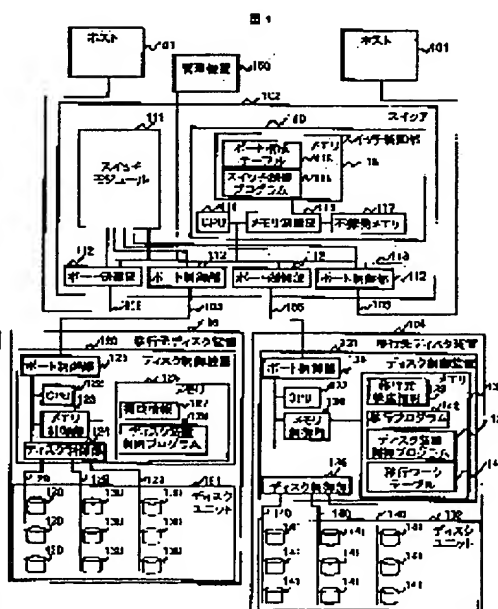
(72)Inventor : WATANABE NAOKI  
TAKAMOTO YOSHIFUMI

## (54) DATA MIGRATION METHOD

## (57)Abstract:

**PROBLEM TO BE SOLVED:** To provide a data migration method suitable for SAN(Storage Area Network) environment and its device.

**SOLUTION:** A disk device at migration destination is connected with a switch in which a host is connected with a disk device at a migration origin. AT this memory, the same value as that of a port ID provided to an F-Port of the switch is provided to a port ID of the disk device at the migration destination and the disk device at the migration destination is made not to be recognized by the host 101. The disk information at the migration destination reads the constitutional information on the disk device at the migration origin. After that, a physical port ID 602 of the disk device 103 at the migration origin is replaced with the physical port ID 602 of the disk device 104 at the migration destination in the switch. The disk device at the migration destination constructs volume according to a logical volume number on the disk device at the migration original and the size of the volume and sores data in the disk device at the migration origin in corresponding volume.



## LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision  
of rejection]

[Date of requesting appeal against examiner's  
decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office



(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2001-249853

(P2001-249853A)

(43) 公開日 平成13年9月14日 (2001.9.14)

(51) Int.Cl. <sup>7</sup>	識別記号	F I	テーマコード* (参考)
G 0 6 F 12/16	3 1 0	G 0 6 F 12/16	3 1 0 M 5 B 0 1 4
3/06	3 0 4	3/06	3 0 4 F 5 B 0 1 8
12/00	5 0 1	12/00	5 0 1 B 5 B 0 6 5
	5 1 1		5 1 1 Z 5 B 0 8 2
13/14	3 1 0	13/14	3 1 0 F 5 D 0 4 4

審査請求 未請求 請求項の数12 O L (全 24 頁) 最終頁に続く

(21) 出願番号 特願2000-63289(P2000-63289)

(22) 出願日 平成12年3月3日 (2000.3.3)

(71) 出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72) 発明者 渡邊 直企

東京都国分寺市東恋ヶ窪一丁目280番地

株式会社日立製作所中央研究所内

(72) 発明者 高本 良史

東京都国分寺市東恋ヶ窪一丁目280番地

株式会社日立製作所中央研究所内

(74) 代理人 100075096

弁理士 作田 康夫

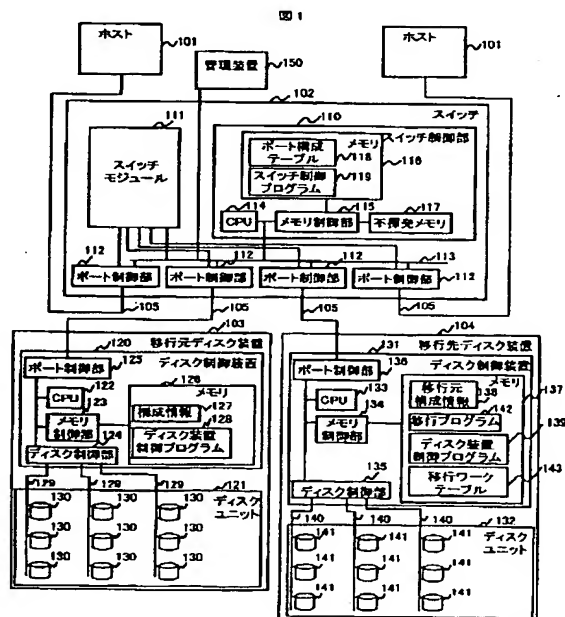
最終頁に続く

(54) 【発明の名称】 データ移行方法

## (57) 【要約】

【課題】 SAN (Storage Area Network) 環境に好適なデータ移行方法、及びその装置を提供することである。

【解決手段】 ホストと移行元ディスク装置が接続されているスイッチに、移行先ディスク装置を接続する。この時点では、移行先ディスク装置のポートIDに、スイッチのF\_Portに与えられたポートIDと同じ値を与えておき、移行先ディスク装置はホスト101から認識されないようにしておく。移行先ディスク情報は移行元ディスク装置の構成情報を読み出す。その後、スイッチにおいて、移行元ディスク装置103の物理ポートID 602と移行先ディスク装置104の物理ポートID 602とを入れ替える。移行先ディスク装置は、移行元ディスク装置上の論理ボリューム番号とそのボリュームの大きさに応じたボリュームを構築し、移行元ディスク装置内のデータを対応するボリュームに格納する。



## 【特許請求の範囲】

【請求項1】スイッチと通信チャネルにより接続され、上記スイッチと通信チャネルにより接続されるホストコンピュータからのリード/ライト要求を受け付ける第1のストレージサブシステムから、第2のストレージサブシステムにデータを移行する方法であって、上記第2のストレージサブシステムを通信チャネルにより上記スイッチと接続し、上記スイッチを介して、上記第1のストレージサブシステムに格納されているデータを上記第2のストレージサブシステムに書き込むことを特徴とするデータ移行方法。

【請求項2】スイッチと通信チャネルにより接続され、上記スイッチと通信チャネルにより接続されるホストコンピュータからのリード/ライト要求を受け付ける第1のストレージサブシステムから、第2のストレージサブシステムにデータを移行する方法であって、上記第2のストレージサブシステムを通信チャネルにより上記スイッチと接続し、上記第1のストレージサブシステムに構築されている論理ボリュームの数、及びそれらのサイズを、上記スイッチを介して、上記第2のストレージサブシステムに読み出し、上記第1のストレージサブシステムに構築されている論理ボリュームと同数及び同サイズの論理ボリュームを上記第2のストレージサブシステムに構築し、ボリューム単位で、上記第1のストレージサブシステムに格納されているデータを、上記スイッチを介して上記第2のストレージサブシステムに書き込むことを特徴とするデータ移行方法。

【請求項3】それぞれポート識別子が割り当てられている複数のポートを有するスイッチの一つのポートと通信チャネルにより接続される第1のストレージサブシステムであって、上記スイッチの他のポートと通信チャネルにより接続されるホストコンピュータから、上記スイッチを介してリード/ライト要求を受け付ける上記第1のストレージサブシステムから、第2のストレージにデータを移行する方法であって、上記第2のストレージサブシステムを、上記スイッチの上記第1のディスク装置及び上記ホストコンピュータが接続されているポート以外のポートに通信チャネルにより接続し、上記第1のストレージサブシステムが接続されているポートに割り当てられているポート識別子と、上記第2のストレージサブシステムが接続されているポートに割り当てられているポート識別子とを互いに入れ替え、上記第1のストレージサブシステム内のデータを、上記スイッチを介して上記第2のストレージサブシステムに書き込み、上記ホストコンピュータから上記第1のストレージサブシステムに対するリード又はライト要求があった場合、その要求は上記第2のストレージサブシステムに送信され、そのリード又はライト要求があったデータが既に上記第2のストレージサブシステムに書き込まれている場合には、上記第2のストレージサブシステムにより、その書き込

まれているデータに対してリード又はライト処理が行われ、そのリード又はライト要求があったデータがまだ上記第2のストレージサブシステムに書き込まれていない場合には、そのデータが上記第2のストレージサブシステムに書き込まれ、上記第2のストレージサブシステムにより、その書き込まれているデータに対してリード又はライト処理が行われることを特徴とするデータ移行方法。

【請求項4】請求項3に記載のデータ移行方法であって、上記第1のストレージサブシステムと上記第2のストレージサブシステムとの間を通信チャネルにより直接接続し、上記第1のストレージサブシステム内のデータを、上記スイッチを介してではなく、上記第1のストレージサブシステムと上記第2のストレージサブシステムとの間を直接接続する通信チャネルを介して、上記第2のストレージサブシステムに書き込むことを特徴とするデータ移行方法。

【請求項5】それぞれポート識別子が割り当てられている複数のポートを有するスイッチの一つのポートと通信チャネルにより接続される第1のストレージサブシステムであって、上記スイッチの他のポートと通信チャネルにより接続されるホストコンピュータから、上記スイッチを介してリード/ライト要求を受け付ける上記第1のストレージサブシステムから、第2のストレージにデータを移行する方法であって、上記第2のストレージサブシステムを、上記スイッチの上記第1のディスク装置及び上記ホストコンピュータが接続されているポート以外のポートに通信チャネルにより接続し、上記第1のストレージサブシステムが接続されているポートに割り当てられているポート識別子と、上記第2のストレージサブシステムが接続されているポートに割り当てられているポート識別子とを互いに入れ替え、上記第1のストレージサブシステムに構築されている論理ボリューム情報を読み出し、上記第2のストレージサブシステムに上記第1のストレージサブシステムに構築されている論理ボリュームと同数及び同サイズの論理ボリュームを構築し、上記第1のストレージサブシステム内のデータを、上記スイッチを介して上記第2のストレージサブシステムに書き込み、上記第1のストレージサブシステムから上記第2のストレージサブシステムへのデータの書き込みが完了したか否かをボリューム単位で管理し、上記ホストコンピュータから上記第1のストレージサブシステムに対するリード又はライト要求があった場合、その要求は上記第2のストレージサブシステムに送信され、そのリード又はライト要求があったデータが既に上記第2のストレージサブシステムへの書き込みが完了している場合には、上記第2のストレージサブシステムにより、その書き込まれているデータに対してリード又はライト処理が行われ、そのリード又はライト要求があったデータがまだ上記第2のストレージサブシステムへの書き込みが

完了していない場合には、そのデータが上記第2のストレージサブシステムに書き込まれ、上記第2のストレージサブシステムにより、その書き込まれているデータに対してリード又はライト処理が行われることを特徴とするデータ移行方法。

【請求項6】請求項5に記載のデータ移行方法であって、上記第2のストレージサブシステムが、上記スイッチを介して上記第1のストレージサブシステムから上記論理ボリューム情報を読み出し、上記第2のストレージサブシステムへのデータの書き込みが完了したか否かをボリューム単位で管理することを特徴とするデータ移行方法。

【請求項7】請求項5に記載のデータ移行方法であって、上記スイッチが、上記第1のストレージサブシステムから上記論理ボリューム情報を読み出し、上記第2のストレージサブシステムに上記第1のストレージサブシステムに構築されている論理ボリュームと同数及び同サイズの論理ボリュームを構築し、上記第2のストレージサブシステムへのデータの書き込みが完了したか否かをボリューム単位で管理することを特徴とするデータ移行方法。

【請求項8】請求項5に記載のデータ移行方法であって、上記スイッチに接続された情報処理装置が、上記第1のストレージサブシステムから上記論理ボリューム情報を読み出し、上記第2のストレージサブシステムに上記第1のストレージサブシステムに構築されている論理ボリュームと同数及び同サイズの論理ボリュームを構築し、上記第2のストレージサブシステムへのデータの書き込みが完了したか否かをボリューム単位で管理することを特徴とするデータ移行方法。

【請求項9】それぞれポート識別子が割り当てられている複数のポートを有するスイッチの一つのポートと通信チャネルにより接続される第1のストレージサブシステムであって、上記スイッチの他のポートと通信チャネルにより接続されるホストコンピュータから、上記スイッチを介してリード／ライト要求を受け付ける上記第1のストレージサブシステムから、第2のストレージにデータを移行する方法であって、上記第2のストレージサブシステムを、上記スイッチの上記第1のディスク装置及び上記ホストコンピュータが接続されているポート以外のポートに通信チャネルにより接続し、上記ホストコンピュータからのリード／ライト要求先を上記第1のストレージサブシステムから上記第2のストレージサブシステムに切り替え、上記第1のストレージサブシステム内のデータを、上記スイッチを介して上記第2のストレージサブシステムに書き込み、上記ホストコンピュータからリード又はライト要求があったデータが既に上記第2のストレージサブシステムに書き込まれている場合には、上記第2のストレージサブシステムにより、その書き込まれているデータに対してリード又はライト処理が

行われ、そのリード又はライト要求があったデータがまだ上記第2のストレージサブシステムに書き込まれていない場合には、そのデータが上記第2のストレージサブシステムに書き込まれ、上記第2のストレージサブシステムにより、その書き込まれているデータに対してリード又はライト処理が行われることを特徴とするデータ移行方法。

【請求項10】請求項9に記載のデータ移行方法であって、上記第1のストレージサブシステムに構築されている論理ボリュームの数、及びそれらのサイズを、上記スイッチを介して、上記第2のストレージサブシステムに読み出し、上記第1のストレージサブシステムに構築されている論理ボリュームと同数及び同サイズの論理ボリュームを上記第2のストレージサブシステムに構築し、ボリューム単位で、上記第1のストレージサブシステムに格納されているデータを、上記スイッチを介して上記第2のストレージサブシステムに書き込むことを特徴とするデータ移行方法。

【請求項11】請求項10に記載のデータ移行方法であって、上記第2のストレージサブシステムは、上記第2のストレージサブシステムへのデータの書き込みが完了したか否かをボリューム単位で管理することを特徴とするデータ移行方法。

【請求項12】請求項1乃至請求項11の何れかに記載のデータ移行方法であって、上記通信チャネルはファイバチャネルであることを特徴とするデータ移行方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、ストレージエリアネットワーク（以下「SAN」という。）に新ディスク装置（移行先ディスク装置）を追加接続し、SANに既に接続されている旧ディスク装置（移行元ディスク装置）からその新ディスク装置にデータを移行する方法及びその装置に関する。

【0002】

【従来の技術】従来のコンピュータシステムでは、複数のサーバがLAN等のネットワークで接続され、各コンピュータには、それぞれ、ディスク装置が直接接続される構成となっていた。各ディスク装置内のデータは、それが直接接続されるサーバにより管理されていた。つまり、データは分散して管理されていた。

【0003】しかし、最近では、複数のサーバ、複数のディスク装置、バックアップ装置等がスイッチ、ハブ等で接続される構成をとるSANがホットなトピックとなっている。これらの装置とスイッチ又はハブ等との間の物理的な接続には、ファイバチャネルが用いられる。SANを用いてシステムを構成するメリットは、スケーラビリティに優れていること、複数のディスク装置に分散したデータを統合して一元管理を行えるため、管理コストの低減を実現できる、といった点にある。そのため、SAN

は、大規模システムの構築に適している。SANでは、多くの資源が相互接続されるため、それらの資源の管理が重要な議題となる。例えば、ディスク装置の管理としては、論理ボリュームの管理（バックアップ等）が挙げられる。現在、それらの管理方法が、SNIA（Storage Networking Industry Association）、NSIC（National Storage Industry Consortium）において検討されている。

【0004】一方、米国特許第5,680,640号には、第1のデータストレージシステムから第2のデータストレージシステムへデータを移行するシステム及びその方法が開示されている。その概要は次の通りである。ホスト、ネットワーク、又はその他のデータ処理システムにあらかじめ接続されている第1のデータストレージシステムが、上記ホストから切り離され、第2のデータストレージシステムに接続される。上記第2のデータストレージシステムは、上記ホスト、又は上記データ処理システムに接続される。上記第2のデータストレージシステムは、どのデータ要素が上記第2のストレージシステムに格納されているか、そして、上記第1のデータストレージシステム上のどのデータが上記第2のストレージシステムにコピーされているかを示すデータマップ、又はデータテーブルを有している。上記ホスト、上記ネットワーク、又は上記データ処理システムが上記第2のストレージシステムにデータを要求すると、上記第2のデータストレージシステムは、そのデータが上記第2のデータストレージシステム、又は上記第1のストレージシステムに格納されているか否かを判断する。もし、データが上記第2のデータストレージシステムに格納されていれば、上記ホスト等に対して、そのデータを使用可能にする。もし、そのデータが上記第2のデータストレージシステムに格納されていない場合には、上記第2のデータストレージシステムは上記第1のデータストレージシステムに要求を発行し、上記ホスト等に対して、そのデータを使用可能にする。そして、上記第2のデータストレージシステムは、そのデータを自分のシステムに書き込み、上記データマップ又は上記データテーブルを更新する。上記第2のデータストレージシステムはビジーでないとき、上記データマップ又は上記データテーブルをスキャンし、上記第1のデータストレージシステムのどのデータが、まだ自分のシステムにコピーされていないかを判断し、そのデータのコピーと、上記データマップ又は上記データテーブルの更新を実行する。

【0005】

【発明が解決しようとする課題】米国特許第5,680,640号に開示されるデータ移行システム及びその方法によれば、上記ホスト等と独立して、上記第1のデータストレージシステムから上記第2のデータストレージシステムへのデータ移行を実行することができる。

【0006】しかし、上記データ移行システムおよびそ

の方法では、上記第2のデータストレージシステムの導入時に、上記第1のデータストレージシステムを上記ホストから切り離し、上記第2のデータストレージシステムに接続し、上記第2のデータストレージシステムを上記ホストに接続するという手続が必要となる。そのため、少なくとも上記第1のデータストレージシステムが上記ホストから切り離されてから、上記第2のデータストレージシステムが上記ホスト及び上記第1のストレージシステムに接続されるまでの間は、上記ホストはI/O（Input/Output）要求を発行することができない。また、その間、上記第1のデータストレージシステムを使用する上記ホスト上のアプリケーション等も一時的に停止される必要がある。データ移行に伴うコストをより抑えるためには、そのI/Oを発行できない時間、アプリケーションを停止する時間をより短縮する必要がある。

【0007】また、上記データ移行システムおよびその方法では、上記第1のデータストレージシステムと上記第2のデータストレージシステムとを接続するために、それらのデータストレージシステムの各々に、チャンネルを新たに設けなければならない。

【0008】さらに、米国特許第5,680,640号には、SAN環境下のデータ移行システム及びその方法が開示されていない。

【0009】本発明の目的は、SAN環境に適したデータ移行システム及びその方法を提供することにある。

【0010】

【課題を解決するための手段】本発明の代表的なデータ移行の概要は次の通りである。

【0011】ホストコンピュータと第1のディスク装置とがそれぞれ通信チャンネルによりスイッチのポートに接続されており、上記第1のディスク装置は上記スイッチを介して上記ホストコンピュータからリード/ライト要求を受け付けているものとする。上記スイッチの各ポートには物理ポートIDと論理ポートIDとが割り当てられており、上記スイッチは、物理ポートIDと論理ポートIDとの対応関係を保持するテーブルを有している。第2のディスク装置を上記ホストコンピュータが接続されているポート及び上記第1のディスク装置が接続されているポート以外の上記スイッチのポートに接続する。上記第2のディスク装置は、上記スイッチを介して、上記第1のディスク装置の構成情報（例えば、論理ボリューム数、各論理ボリュームのサイズ）を取得する。その後、上記第1のディスク装置が接続されるスイッチのポートに割り当てられている物理ポートIDと論理ポートIDとの対応関係と、上記第2のディスク装置が接続されるスイッチのポートに割り当てられている物理ポートIDと論理ポートIDとの対応関係とを入れ替える。具体的には、上記第1のディスク装置が接続されるスイッチのポートに割り当てられている論理ポートIDと上記第2のディスク装置が接続されるスイッチのポートに割り当てられて

いる論理ポートIDとを入れ替える。これにより、ホストコンピュータから上記第1のディスク装置にアクセスしても、実際には、上記第2のディスク装置にアクセスされることになる。このポートID切替処理を行った後、上記第2のディスク装置上に、上記第1のディスク上の構成情報に対応する論理ボリュームが構築され、上記第1のディスク装置内のデータが上記第2のディスク装置に移行される。上記ホストコンピュータから、上記第2のディスク装置に既に移行されているデータに対してリード又はライト要求が合った場合には、上記第2のディスク装置によりそのデータに対してその処理が行われる。上記ホストコンピュータから、上記第2のディスク装置にまだ移行されていないデータに対してリード又はライト要求が合った場合には、そのデータが上記第1のディスク装置から上記第2のディスク装置に読み出され、上記第2のディスク装置によりそのデータに対してその処理が行われる。

【0012】その他の本願が提供するデータ移行方法は、発明の実施の形態の欄で明らかにされる。

【0013】

【発明の実施の形態】以下、実施例1乃至実施例5は、SAN環境に好適なデータ移行方法及びその装置を提供する。

【0014】〔実施例1〕本実施例では、ホストと移行元ディスク装置が接続されているスイッチに、移行先ディスク装置が接続され、そのスイッチを用いて、移行元ディスク装置内のデータが、移行先ディスク装置に移行される。従って、本実施例によれば、ホストと移行元ディスク装置との間の接続を切り離し、移行先ディスク装置をホスト及び移行元ディスクに接続するという作業は必要なく、単に、移行先ディスク装置をスイッチに接続するだけなので、上述の米国特許第5,680,640号に開示されるデータ移行システム及びその方法に比べて、移行先ディスク装置を追加する作業が短縮される。そのため、データ移行に伴うコストもより抑えることができる。また、本実施例によれば、データ移行のための専用チャンネルを設ける必要もないので、ハードウェアのコストも、より抑えることができる。

【0015】以下、図1から図10を用いて、本実施例を説明する。

【0016】(1) システム構成

図1は、本発明のデータ移行システム及びその方法が適用されるコンピュータシステムの第1の実施例を説明するための図である。2つのホスト101、移行元ディスク装置103、及び移行先ディスク装置104が、スイッチ102により接続されており、これらがSANを形成している。ホスト101、移行元ディスク装置103、及び移行先ディスク装置104とスイッチ102との間はファイバチャネル105で接続されている。なお、本発明は、移行元ディスク装置103から移行先ディスク装置104へデータ移行するとい

うものであるから、本発明を実施する上で、ホストの台数は制限されない。

【0017】図5は、ファイバチャネル105において使用されるフレームの構造を示す。フレームはフレームの先頭を示すSOF (Start Of Frame) 501と、終わりを示すEOF (End Of Frame) 505との間にFRAME HEADER 502、DATA FIELD 503とCRC 504 (Cyclic Redundancy Check) が入る。FRAME HEADER 502はフレームの制御情報を含み、受信側アドレスであるD\_ID (Destination ID) 507と送信元アドレスであるS\_ID (Source ID) 508、ルーティングの制御を行うR\_CTL (Routing Control) 506、データの構造を示すTYPE 510、フレームのシーケンスやエクスチェンジの制御を行うF\_CTL (Frame Control) 511、送信元と受信側のシーケンスの識別をおこなうSEQ\_ID (Sequence ID) 512、各シーケンス毎のフレームの数のカウント値を示すSEQ\_CNT (Sequence count) 514、データフィールドの制御情報であるDF\_CTL (Data Field Control) 513を含む。ファイバチャネル105のフレームについてはANSIのX3.230 FC-PH (Fibre Channel Physical and Signaling Interface) に詳しい。

【0018】図3は、図1に示されるコンピュータシステムの論理的なネットワーク接続構成を示す。そのコンピュータシステムでは、ファブリック301は一つのスイッチ102で構成されている。ホスト101、ディスク装置との接続に使用するスイッチ102のポート303は、F (Fabric) ポートと呼ばれる。また、ホスト101、ディスク装置のポート302は、N (Node) ポートと呼ばれる。図4に示すように、ファブリック401は、複数のスイッチ102で構成されることも可能である。スイッチ間を接続するスイッチのポートはE (Expansion) ポート401と呼ばれる。

【0019】図3のようにファブリックが一つのスイッチ102で構成されている場合、ホスト101から移行先ディスク装置104へのデータ転送は次のようにして行われる。ホスト101は、自分のポートのIDをS\_ID 508に格納し、フレームの送信先のポートIDをD\_ID 507に格納しフレームを送信する。スイッチ102は、フレームを受信すると、D\_ID 507に格納されているポートIDを調べ、そのポートIDと一致するF\_Portから、移行先ディスク装置104のN\_Portへフレームを送信する。S\_ID 508及びD\_IDは論理的なポートIDであり、また、図4に示したように、ファブリック401が複数のスイッチ102で構成されている場合、ホスト101から移行先ディスク装置104へのデータ転送は次のようにして行われる。ホスト101と直接接続されているスイッチ102は、ホスト101からフレームを受信すると、D\_ID 507に格納されているポートIDを調べる。しかし、そのスイッチは、そのポートIDと一致するF\_Portを持っていないので、E\_Portから他のスイッチにフレームを送信する。そして、そのポートIDと一致するF\_Portを持っている移行先ディスク装置104に直接接続されているスイッチが、そのF\_Portから移行先ディスク

装置104のN\_Portへフレームを送信する。以上、ホスト101から移行先ディスク装置104へのデータ転送を例にとり説明したが、いずれの装置間のデータ転送も同様に行われる。

【0020】再び、図1を参照して、スイッチ102、移行元ディスク装置103、移行先ディスク装置104、及びホスト101の順で、それらの構成を説明する。

【0021】(A) スイッチ102の構成  
スイッチ102は、スイッチ制御部110、スイッチモジュール111、及びポート制御部112を含む。

【0022】スイッチ制御部110は、スイッチ102内の制御を行うものであり、CPU114、メモリ制御部115、メモリ116、及び不揮発メモリ117を含む。

【0023】CPU114は、内部バス113を用いて、各制御部と制御情報及びデータのやり取りを行う。

【0024】不揮発メモリ117は、メモリ制御部115によりリード/ライトが制御され、スイッチ102の制御に必要なスイッチ制御プログラム119、及びそのプログラムの実行時に必要なポート構成テーブル118等のデータを格納する。

【0025】図6は、ポート構成テーブル118の構成例を示す図である。図6(A)は、ポート切替処理前のポート構成テーブルを示しており、図6(B)は、そのポート切替処理後のポート構成テーブルを示している。ポート切替処理については後述する。ポート構成テーブル118は、ポートの構成を示すものであり、論理的なポートIDを示す論理ポートID 601、及び物理的なポートIDを示す物理ポートID 602を含む。その他に、ファイバチャネルの転送サービスクラスの規定、ループ等のポートの種別が含まれていてもよい。本実施例では、図20に示すように、論理ポートID\_0にホスト101が、論理ポートID\_1に移行元ディスク装置103が、論理ポートID\_4にホスト101が接続されているものとする。また、論理ポートID\_2に移行先ディスク装置104が接続されるものとする。

【0026】メモリ116は、メモリ制御部115によりリード/ライトが制御される。スイッチ制御プログラム119がCPU114により実行されるとき、そのプログラムが不揮発メモリ117から読み出され、メモリ116上に格納される。また、その際、ポート構成テーブル118も、必要に応じてCPU114が不揮発メモリ117から読み出され、メモリ116上に格納される。

【0027】ポート制御部112は、データの符号化/複合化等の制御を行う。具体的には、ポート制御部112は、受信したファイバチャネルのフレームからシリアルデータを取り出してパラレルデータに変換し、また、そのフレームから送信先ID等、スイッチングに必要な情報を取り出してスイッチモジュール111に送信する。また、ポート制御部112は、その逆の処理も行う。

【0028】スイッチモジュール111は複数のポートを有しており、それらは、それぞれ複数のポート制御部11

2に接続される。スイッチモジュール111は、内部バス113を介してCPU114と接続されており、CPU114により制御される。スイッチモジュール111は、ポート制御部112からデータを受信すると、そのデータの送信先IDに従い、出力すべきポートを切り替え、データを送信する。スイッチモジュール111はクロスバススイッチ等で構成すればよい。

【0029】スイッチ102に接続される制御装置150は、スイッチ102の各種パラメータの設定、本実施例の最も特徴でべきな機能である移行処理の制御及び情報管理を行う。また、管理装置150は、後述する移行先ディスク装置104内の移行プログラムを起動させるためのコマンド、移行情報（例えば移行中、移行終了、異常発生）を取得するためのコマンドを持っている。

【0030】なお、図1では、管理装置150はスイッチ102に直接接続されているが、図22に示すように、ホスト101、スイッチ102、移行元ディスク装置103、移行先ディスク装置104、及び制御装置150をLAN (Local Area Network) に接続し、Web等により上記の設定を行うようにしてもよい。

【0031】(B) 移行元ディスク装置102及び移行先ディスク装置103の構成

本実施例では、説明を簡単にするため、移行元ディスク装置103と移行先ディスク装置104とは、メモリ上のプログラム及び情報以外は同様の構成とする。

【0032】移行元ディスク装置103は、ディスク制御装置120とディスクユニット121とを含む。

【0033】ディスクユニット121は、複数のディスクドライブ130を含んでいる。それらのディスクドライブ130はインターフェース (I/F) 129 (ファイバチャネル、SCSI等) によりディスク制御部124に接続されている。しかし、本発明を実施する上では、ディスクドライブ130の台数は制限されない。

【0034】ディスク制御装置120は、CPU 122、メモリ制御部123、ポート制御部125、ディスク制御部124、及びメモリ126を含む。CPU 122、メモリ制御部123、ポート制御部125、及びディスク制御部124とは内部バスで接続されている。

【0035】CPU 122は、それらの制御部と、その内部バスを用いて制御情報、データ等のやり取りを行う。CPU 122は、スイッチ102経由でホスト101から送信されたコマンドに応じて、そのコマンドの処理に必要なディスクドライブ130にリード・ライト命令を発行する。CPU 122は、複数のディスクドライブ130により、よく知られているRAID 0~5の構成を構築し、ホスト101に対して論理的なボリュームを提供する。

【0036】ポート制御部125は、ファイバチャネル105により、スイッチ102のポート制御部112に接続されており、その機能はポート制御部112と同様である。

【0037】メモリ126は、メモリ制御部123に接続され

ており、ディスク装置を制御するディスク装置制御プログラム128、該プログラムの実行時に必要なデータ、及び構成情報127を格納する。構成情報127は後述される。ディスク装置制御プログラム128は、CPU 122により実行され、ポート制御部125及びディスク制御部124の制御、及びホスト101から受信したリード／ライトコマンドの処理を行うプログラムである。

【0038】上述のように、移行先ディスク装置104の構成要素は、メモリ137上のプログラム及び情報以外は、移行元ディスク装置103の構成要素と同様であるので、メモリ137上のプログラム及び情報のみ説明する。メモリ137上には、移行元ディスク装置103のメモリ126に格納されている構成情報127が読み取られ、それが入力されたものである移行元構成情報138と、データの移行時に使用する移行プログラム142、データの移行の状態を示す移行ワークテーブル143が格納される。

【0039】図8は、移行ワークテーブル143の構成例を示す。移行ワークテーブル143は、ボリュームの番号801、スロットの番号802、及び該スロットの状態を表すステータス803を含む。

#### 【0040】(C) ホスト101の構成

図2は、ホスト101の構成を示す。ホスト101は、CPU 201、メモリ制御部202、ポート制御部206、ディスク制御部204、ディスクドライブ205、及びメモリ203を含む。

【0041】CPU 201、メモリ制御部202、ポート制御部206、ディスク制御部204とは内部バス207で接続される。CPU 201は、内部バス207を用いて、それらの制御部と、制御情報及びデータのやり取りを行う。

【0042】ポート制御部206は、ファイバチャネル105を介してスイッチ102のポート制御部112と接続されており、そのポート制御部102とコマンド、データ等のやり取りを行う。

【0043】ディスクドライブ205は、ディスク制御部204に接続されオペレーティングシステム208、ポート制御部206等のハードウェアを制御するデバイスドライバ209、アプリケーションプログラム210、及びこれらのプログラムの実行時に必要なデータを格納する。

【0044】メモリ203は、メモリ制御部202と接続される。オペレーティングシステム208、ディスク制御部204、及びデバイスドライバ209、アプリケーションプログラム210等がCPU210により実行される時、ディスクドライブ205から読み出され、メモリ203上に格納される。

#### 【0045】(2) データ移行処理のフロー

次に、図9を参照して、本発明のデータ移行処理を説明する。なお、ホスト101と移行元ディスク装置103とは、既にスイッチ102に接続されているものとする。

【0046】(A) 移行先ディスク装置をスイッチに接続 (901)

まず、オペレータは、制御装置150を起動し、移行元デ

ィスク装置103及び移行先ディスク装置104が接続されているスイッチ102のポート番号を入力する。

【0047】次に、オペレータは、移行先ディスク装置104をスイッチ102に接続する。この時点では、移行先ディスク装置104のポートIDには、既に使用されているポートID以外のポートIDであればどのようなポートIDを付けても良い。スイッチ102の各ポートにデフォルトでポートIDが割り当てられているときは、移行先ディスク装置104が接続されるF\_Portに与えられているポートIDを与えればよい。本実施例では、上述のように、移行先ディスク装置104には、論理ポートID<sub>2</sub>が与えられる。

【0048】移行先ディスク装置104に割り当てられた論理ポートIDは、ホスト101が使用できるディスク装置の論理ポートID<sub>1</sub>とは異なるので、この時点では、ホスト101は、移行先ディスク装置104にアクセスすることはできない。

【0049】(B) 移行元ディスク装置の構成情報を移行先ディスク装置に入力 (902)

移行先ディスク装置104の接続後、オペレータは、管理装置150から上述のコマンドにより、移行先ディスク装置104内の移行プログラム142を起動させる。起動された移行プログラム142は、まず、移行元ディスク装置103から構成情報127を取得する (902)。

【0050】図7に示すように、構成情報127は、移行元ポートID 702、World wide Name 703、ファイバ構成情報704、SCSI構成情報705を含む。

【0051】ファイバ構成情報704は、PLOGIペイロード706とPRLIペイロード707を含む。PRLIペイロード707は、ファイバチャネルのN\_Port Login (PLOGI) の際にやり取りされる共通サービス・パラメータ708、ポート名709、ノード名710、及びクラス・サービス・パラメータ711を含む。共通サービス・パラメータ708には、ファイバチャネルのバージョン情報や該ファイバチャネル機器がサポートしているアドレス指定方法等の機能、及び通信方法等が指定されている。クラス・サービス・パラメータ711は、クラスのサポート情報、X\_IDの再割当、ACKの能力等を示す。ファイバチャネルのパラメータに関しては、上述のANSIのX3.230 FC-PHに詳しく記載されている。PRLIペイロード707は、サービス・パラメータ712を含む。

【0052】SCSI構成情報705は、INQUIRYデータ713、ディスコネクト・リコネクト・パラメータ714、センスデータ715、及びモード・セレクト・パラメータ716を含む。INQUIRYデータ713は、SCSI装置のタイプ、ベンダID、プロダクトID等を示す。ディスコネクト・リコネクト・パラメータ714は、接続の条件を示す。センスデータ715は、エラー発生時にディスク装置の状態を調べるためにやり取りされるものである。モード・セレクト・パラメータ716は、SCSI装置の物理的属性、記憶媒体上のデータ形式、エラーリカバリの方法や手順、I/Oプロ



セスの処理方法などに関する各種のパラメータの設定や変更を行うものである。INQUIRYデータ713とモード・セレクト・パラメータ716とにより、移行元ディスク装置103内のポリウム数、各ポリウムのサイズ（ブロック数）を知ることができる。ファイバチャネル105上のSCSIの詳細なプロトコルに関しては、ANSIのX3.269 Fibre Channel Protocol for SCSIに詳しい。

【0053】構成情報127は上記のように既存のファイバチャネル、SCSI等のプロトコルを用いて取得できる情報以外にも存在し得る。そのような構成情報が存在する場合には、オペレータが、移行元ディスク装置103から構成情報127を直接読み取り、移行先ディスク装置104に入力する。読み取り、入力は、操作パネル、又はHTTPによるwebアクセスにて取得する。

【0054】移行先ディスク装置104の移行プログラム142は、移行先ディスク装置104への構成情報127の入力、すなわち、移行元構成情報138の入力が終了すると、その旨をスイッチ102に通知する。

【0055】(C) ポート切替処理 (903)

スイッチ102のスイッチ制御プログラム119が、その通知を受けると、ポート切換処理が開始される。

【0056】ポート切換処理の詳細を、図10のフローチャートを用いて説明する。

【0057】最初に図3に示したように、ファブリック301が一つのスイッチ102で構成される場合について説明する。

【0058】まず、スイッチ102のスイッチ制御プログラム119が、移行元ディスク装置103を使用している全ホスト101に、ポート切換を開始する旨を通知する。その通知を受け取ったホスト101のデバイスドライバ209は、移行元ディスク装置103に対するI/Oを、ホスト101のメモリ203上にキューイングする（1001）。

【0059】移行元ディスク装置103へのI/Oが停止されると、デバイスドライバ209は、スイッチ102に対してI/Oの停止完了を通知する。移行元ディスク装置103に対して実行中のI/O処理がある場合は、中止させてもよいが、終了するまで実行させた後、I/Oの停止完了が通知されるのが望ましい。スイッチ102のスイッチ制御プログラム119は、全ホスト101からその通知を受け付けた後、ポート構成テーブル118の移行元ディスク装置103の論理ポートID 601と物理ポートID 602との対応関係、移行先ディスク装置104の論理ポートID 601と物理ポートID 602との対応関係を変更する（1002）。つまり、ポート構成テーブル118は、図6（B）のように書き換える。この様子をコンピュータシステム全体の図で表すと、図20に示す状態から図21に示す状態に変化することとなる。

【0060】以後、スイッチ102のポート制御部112は、フレームの送受信毎にポート構成テーブル118を参照することにより、S\_ID 508及びD\_ID 507を操作し、ポートの切換処理を行う。フレームの受信時には、受信したフ

レームのS\_ID 508に対応する論理ポートID 601が検索され、そのフレームのS\_ID 508は、その検索された論理ポートID 601に対応する物理ポートID 602に変換される。同様に、フレーム送信時は、フレームのD\_ID 507と対応する論理ポートID 601が検索され、そのD\_ID 507は、その検索された論理ポートID 601に対応する物理ポートID 602に変換される。このとき、フレームに付随しているCRC 504は計算し直される。以上の処理により、移行元ディスク装置103に対するフレームは、すべて移行先ディスク装置104に送信される。また、移行先ディスク装置104から送信されたフレームは、ホスト101からは、移行元ディスク装置103から送信されたように見える。スイッチ102内でのポート切換後、ホスト101のI/Oを再開する。（1003）次に、図4に示したように、ファブリック401が複数のスイッチ102で構成される場合について説明する。

【0061】まず、マスタとなるスイッチ102（マスタスイッチ）を決める。本実施例では移行先ディスク装置104が直接接続されているスイッチ102が、マスタスイッチとなる。

【0062】マスタスイッチのスイッチ制御プログラム119は、移行元ディスク装置103を使用している全ホスト101、及びファブリック401内のマスタスイッチ以外の全スイッチ102に、ポート切換を開始する旨を通知する。その通知を受け取ったホスト101のデバイスドライバ209が行う処理は、I/Oの停止完了をマスタスイッチに通知することを除いて、ファブリックが一つのスイッチ102で構成されている場合と同じである。

【0063】マスタスイッチのスイッチ制御プログラム118は、全ホスト101からその通知を受け付けた後、ポート構成テーブル118の移行元ディスク装置103の論理ポートID601と物理ポートID 602との対応関係と、移行先ディスク装置104の論理ポートID 601と物理ポートID 602との対応関係を変更する共に、その変更をマスタスイッチ以外の全スイッチ102に通知する。マスタスイッチ以外の全スイッチは、その通知に基づき、自身のポート構成テーブル118を変更する。以後の各スイッチの動作は、ファブリックが一つのスイッチ102で構成されている場合のスイッチ102の動作と同じである。ファブリック401内の全スイッチ102のポート切換が行われた後、ホスト101のI/Oを再開する。

【0064】(D) データ移行処理 (904)

ポート切替処理が終了に同期して、データ移行処理が行われる。図11のフローチャートを用いて、そのデータ移行処理を説明する。

【0065】まず、移行プログラム142は、移行元ディスク装置103上の論理ボリューム番号とそのボリュームの大きさに応じたボリュームを移行先ディスク装置104に構築し、使用する変数及び図8で説明した移行ワークテーブル143の初期化を行う（1101）。



【0066】移行先ディスク装置104上の移行プログラム142は、ホスト101からのI/O要求の有無をチェックする(1102)。

【0067】ホスト101からのI/O要求が無い場合は、移行元ディスク装置103から、スロット単位でデータを移行する。その際、移行プログラム142は、移行ワークテーブル143を用いて、次に移行するデータのアドレスを算出する(1103)。

【0068】移行元ディスク装置103の全論理ボリュームに対して、ボリューム番号801の小さい順からデータ移行が実行され、また、各ボリューム内においては、先頭ブロックからデータ移行が行われる。図8に示した移行用ワークテーブル132は、その先頭行のボリューム番号801の欄に次移行アドレスを示す。次移行アドレスの初期値は、最小のボリューム番号801の先頭ブロックアドレスとなる。また、移行ワークテーブル143は、その先頭行のスロット番号802の欄に次移行スロットを示す。2番目以降の行はボリューム番号801、スロット番号802の昇順となるよう整列される。移行プログラム142は、移行ワークテーブル143から得た次移行アドレスのデータを移行する際、そのスロット番号のステータス803を「移行中」へ変更する(1104)。

【0069】次に移行プログラム142は、ポート制御部125を用いて移行元ディスク装置103へスロットのサイズのリード要求を発行し、該当するデータを取得する(1105)。

【0070】移行プログラム142は、移行元ディスク装置103から得たデータを移行先ディスク装置104の対応するボリュームへディスク制御部124を使用しI/F 140を介しディスクドライブ141へ書き込む(1106)。

【0071】ディスクドライブ141へデータを書き込んだ後、当該ボリュームの全スロットの移行が終了したかチェックし、終了している場合には次のボリュームのデータ移行を実行し、終了していない場合はスロット番号をインクリメントする(1107)。

【0072】全ボリュームに関する移行が終了している場合には移行処理を終了する(1108)。

【0073】次にホスト101からのI/O要求があった場合について説明する。ホスト101からのI/O要求がある場合には、移行プログラム142はリードかライトか調べる(1109)。

【0074】移行プログラム142はそれがリードの場合、移行ワークテーブル143を参照し、要求データが移行済みかどうか調べる(1110)。

【0075】移行プログラム142は、移行済みの場合には、移行先ディスク装置104内のディスクドライブ141から読み出し、ホスト101に返送する(1113、1112)。

【0076】ホスト101から要求されたデータが移行済みでない場合には、移行プログラム142はポート制御部125を用いて、スイッチ102経由で移行元ディスク装置103

にリード要求を発行し、当該データを取得する。移行プログラム142は、移行元ディスク装置103から取得したデータをホスト101へ返送すると共に、移行先ディスク装置104の対応するボリュームへそのデータを書き込む(1111、1112)。

【0077】ディスクドライブ141へ当該データを書き込んだ後、該データを移行ワークテーブル143に登録し、ステータス803を移行済みとし、該要求に対する処理を終える(1108)。

【0078】次に、ホスト101からのライト要求があった場合について説明する。移行プログラム142は、ホスト101からライト要求に関するデータを受け取る(1114)。

【0079】次に、移行プログラム142は、ポート制御部125を用いて、スイッチ102経由で、移行元ディスク装置103へホスト101からの要求データを含むようスロットサイズ単位のリード要求を発行しデータを取得する。移行プログラム142は移行元から得られたデータにホスト101から受け取ったデータを上書きし新しいスロットデータを作成した後、移行先ディスク装置104内のホスト101が要求したボリュームに対して書き込みを行う(1115)。

【0080】書き込みが終了したら、移行プログラム142は移行ワークテーブル143へ該スロットを登録し、ステータス803を「移行済み」とし、ホスト101からのライト要求に関する処理を終える(1107)。

【0081】(E) 移行元ディスク装置の切り離し(905)

データ移行処理の終了後、オペレータは移行元ディスク装置103をスイッチから切り離し、移行処理を終了する(905)。

【0082】以上、実施例1のシステム構成及びデータ移行処理のフローを説明した。本実施例によれば、ホストからのI/O要求が停止されるのは、移行元ディスク装置103の構成情報を移行先ディスク装置104に入力されている間、及びポートの切替処理が行われている間だけである。従って、本実施例によれば、ホストと移行元ディスク装置との間の接続を切り離し、移行先ディスク装置をホスト及び移行元ディスクに接続するという作業は必要ないので、ホストからのI/O要求が停止される時間は、米国特許第5,680,640号に開示されるデータ移行システム及びその方法においてホストからのI/O要求が停止される時間に比べて短い。そのため、データ移行に伴うコストをより抑えることができる。さらに、本実施例によれば、データ移行のための専用チャンネルを設ける必要もないので、ハードウェアのコストも、より抑えることができる。

【0083】〔実施例2〕本実施例は、実施例1で説明したシステム構成において、さらに、移行元ディスク装置と移行先ディスク装置とを、インターフェース(例え

ばファイバチャネル又はSCSI)で直接接続する点で、実施例1と異なる。本実施例では、そのインタフェースを用いてデータ移行を行うので、実施例1に比べて高速にデータ移行を行うことができる。本実施例では、データ移行用のインターフェースを設ける必要がある。しかし、ホストと移行元ディスク装置との間の接続を切り離し、移行先ディスク装置をホスト及び移行元ディスクに接続するという作業は必要なく、単に、移行先ディスク装置をスイッチに接続し、移行元ディスク装置と移行先ディスク装置とをインターフェースで接続するだけなので、上述の米国特許第5,680,640号に開示されるデータ移行システム及びその方法に比べて、移行先ディスク装置を追加する作業が短縮される。

【0084】以下、図12を用いて、本実施例を説明する。

【0085】図12は、本発明のデータ移行システム及びその方法が適用されるコンピュータシステムの第2の実施例を説明するための図である。上述のように、そのコンピュータシステムの各構成要素は、図1の各構成要素と同様であるので、説明は省略する。

【0086】以下、本実施例におけるデータ移行処理フローを説明する。データ移行処理フローは、図9で説明した実施例1のものと同様である。但し、本実施例では移行元ディスク装置1203と移行先ディスク装置1204とが直接インターフェースで接続されており、そのインターフェースを用いてデータ移行を行うので、移行元構成情報1238により、移行元ディスク装置1203内の論理アドレスと物理アドレスの対応関係を求める必要がある。以下、その異なる点のみを説明する。

【0087】図11のステップ1105、1113、及び1115では、スイッチ102を介して移行元ディスク装置からデータを読み込んだ。しかし、本実施例では、移行プログラム1231は、移行元構成情報1238に含まれる情報を用いて、移行元ディスク装置1203から読み出すべきデータの論理アドレスに対応する物理アドレスを算出する。そして、移行プログラム1231は、1/F1240を用いて、算出した物理アドレスに対応する移行元ディスク装置1203のディスクドライブ1230に対して、直接リード要求を発行し、当該データを取得する。このとき、当該データが移行元ディスク装置1203の複数のディスクドライブ1230に分散されている場合、又は離散した物理アドレスに格納されている場合は、複数のリード要求を発行する。

【0088】本実施例では、移行元ディスク装置1203と移行先ディスク装置1204とを直接インタフェースで接続し、スイッチ1202を介さず、そのインターフェースを用いてデータ移行を行うため、実施例1に比べて高速にデータ移行を行うことができる。また、本実施例のデータ移行はスイッチ1202を介さないで、スイッチのリソースの消費を抑えることができる。本実施例では、スイッチ1202に接続されているのは、ホストが2台、ディスク

装置が2台であるが、SAN環境下では、多くのホスト及びディスク装置が接続されるので、そのリソースを他のホストと他のディスク装置間のデータ転送に割り当てる事が可能となる。従って、本実施例によれば、実施例1に比べて、データ移行に伴うスイッチの性能低下を抑制することができる。

【0089】〔実施例3〕本実施例では、移行プログラムをスイッチで実行する点で実施例1と異なる。本実施例によれば、ディスク装置に移行プログラムを持たせる必要がないので、そのようなプログラムを持たないディスク装置でも移行元ディスク装置からのデータ移行が可能となる。

【0090】図13を用いて、本実施例のコンピュータシステムの構成を説明する。但し、図1の構成要素と同様のものについては説明を省略する。

【0091】本実施例では、スイッチ1302内の不揮発メモリ1317に、スイッチ1302の制御に必要なスイッチ制御プログラム1319、移行処理を行う移行プログラム1342と、それらのプログラムの実行時に必要なポート構成テーブル1318、移行元構成情報1327、及び移行ワークテーブル1343等のデータとを格納する。移行先ディスク装置1304内のメモリ1337にはディスク装置制御プログラム1339が格納されるが、実施例1のように移行元構成情報や移行ワークテーブル、移行プログラムは格納されない。

【0092】図14は、ポートの構成を示すポート構成テーブル1318の構成例である。ポート構成テーブル1318は、仮想的なポートIDを示す論理ポートID 1401、物理的なポートIDを示す物理ポートID 1402、及びポートの状態を示すステータス1403を含む。

【0093】次に、図15のフローチャートを用いて、本実施例のデータ移行処理を説明する。

【0094】オペレータは、移行先ディスク装置1304をスイッチ1302に接続する(1501)。このステップは、図9のステップ901と同様である。

【0095】次に、スイッチ1302の移行プログラム1342は、移行元ディスク装置1303から構成情報1327を読み取る(1502)。構成情報1327は図7に示すものと同様である。

【0096】移行元ディスク装置1303の構成情報1327がスイッチ1302に入力された後、スイッチ1302のスイッチ制御プログラム1319はポート切替処理を開始する(1503)。

【0097】本実施例のポート切替処理においても、実施例1と同様に、ホストのI/Oを止めた後、ポート構成テーブル1318の移行元ディスク装置1303の論理ポートID 1401と物理ポートID 1402との対応関係と、移行先ディスク装置1304の論理ポートID 1401と物理ポートID 1402との対応関係を変更する。しかし、本実施例のポート切替処理では、次の点で実施例1のそれと異なる。本実施例では、スイッチ1302のスイッチ制御プログラム1319は、

ポート構成テーブル1318における移行元ディスク装置1303のポートIDのステータス1403を「移行中」とする。ポート構成テーブル1318のステータス1403が「移行中」のポートIDに対するフレームは、対応するポートには送信されず、スイッチ1302の移行プログラム1331に渡される。つまり、移行元ディスク装置1303へのアクセスはスイッチ1302の移行プログラム1331が行う。

【0098】ファブリックが複数のスイッチ1302により構成されている場合には、実施例1と同様のやり方でホストのI/Oを止めた後、マスタスイッチのスイッチ制御プログラム1319は、ポート構成テーブル1318における移行元ディスク装置1303のポートIDのステータス1403を「移行中」とする。ポート構成テーブル1318のステータス1403が「移行中」のポートIDに対するフレームは、対応するポートには送信されず、スイッチ1302の移行プログラム1331に渡される。ファブリックが複数のスイッチ1302で構成されている場合には、少なくともマスタスイッチとなるスイッチ1302に移行元構成情報1327、移行ワークテーブル1343、及び移行プログラム1342が搭載されていれば足りる。

【0099】ポート切替処理が終了した後、データ移行処理を行う。まず、スイッチ1302の移行プログラム1342は、移行元ディスク装置1303上に構成されている論理ボリューム番号と該ボリュームの大きさに従ったボリュームを構築し、使用する変数や図8に示す移行ワークテーブル1332の初期化を行う。以後の処理は、移行先ディスク装置の移行プログラムではなく、スイッチ1302の移行プログラム1302が行うという点を除き、実施例1で説明した処理と同様である。

【0100】データ移行処理が終了した後、スイッチ1302のスイッチ制御プログラム1319は、ポート構成テーブル1318の移行先ディスク装置1304のステータスを「通常」に変更する(1505)。オペレータは移行元ディスク装置1303をスイッチ1302から切り離し、移行処理を終了する(1506)。

【0101】〔実施例4〕本実施例の特徴は、スイッチに接続されるパーソナルコンピュータ、ワークステーション等で構成されるマイグレータ上で移行プログラムが実行される点である。本実施例によれば、実施例3と同様に、移行先ディスク装置に移行プログラムを持たせる必要がないので、そのようなプログラムを持たないディスク装置でも移行元ディスク装置からのデータ移行が可能となる。さらに、本実施例では、実施例3のように、スイッチ上で移行プログラムを実行させないので、スイッチの負荷を減らすことが可能となる。

【0102】図16を用いて、本実施例のコンピュータシステムの構成を説明する。但し、図1の構成要素と同様のものについては説明を省略する。

【0103】本実施例では、後述するようにマイグレータ1606内に、移行処理を行う移行プログラムと、移行元

構成情報、及び移行ワークテーブルを格納する点に特徴がある。移行先ディスク装置1304内のメモリ1337にはディスク装置制御プログラム1339が格納されるが、実施例1のように移行元構成情報や移行ワークテーブル、移行プログラムは格納されない。また、実施例3と異なり、本実施例では、スイッチ1602に、移行プログラムと、移行元構成情報、及び移行ワークテーブルは格納されない。

【0104】図17は、マイグレータ1606の構成例を示す。マイグレータ1606はCPU 1701、メモリ制御部1702、ポート制御部1706、ディスク制御部1704、ディスクドライブ1705、メモリ1703を含む。CPU 1701と、メモリ制御部1702、ポート制御部1706、及びディスク制御部1704とは内部バス1707で接続され、CPU 1701は各制御部との間で制御情報、データのやり取りを行う。ポート制御部1706は、ファイバチャネル1605によりスイッチ1602のポート制御部1612と接続され、スイッチ1602とコマンド、データのやり取りを行う。マイグレータ1606のポートはNポートと呼び、スイッチ1602側のポートをFポートと呼ぶ。ディスクドライブ1705はディスク制御部1704と接続され、CPU 1701からのリード・ライトの指示を受け付ける。ディスクドライブ1705にはマイグレータ1606に必要なプログラムが格納されており、移行プログラム1731及び、該プログラムの実行時に必要な移行ワークテーブル1732等が格納する。メモリ1703はメモリ制御部1702と接続され、移行プログラム1731の実行時に当該プログラムがドライブ1705からメモリ1703上に読み出される。

【0105】次に、本実施例のデータ移行処理のフローを説明する。オペレータは移行先ディスク装置1604をデータ移行用スイッチ1602に接続する。この時には、移行先ディスク装置1604のポートIDはスイッチ1602のF\_Portに与えられたポートIDと同じ値を取り、移行先ディスク装置1604はホスト1601からは認識されない。このステップは、図9で説明したステップ901と同様である。

【0106】次に、マイグレータ1606の移行プログラム1631は、移行元ディスク装置1603から構成情報1727を読み取る。構成情報1727は図7に示すものと同様である。

【0107】移行元ディスク装置1603の構成情報1727がマイグレータに入力された後、ポートの切替処理を行う。その処理は、実施例1で説明したものと同様に、ホストからのI/Oを停止させ、ポート構成テーブル1618の移行元ディスク装置1603の物理ポート1604と移行先ディスク装置1604の物理ポートID1402とを入れ替える。スイッチ1602の制御プログラムはポート構成テーブル1618の移行元ディスク装置1603のポートIDのステータスを「移行中」とする。ポート構成テーブルのステータスが「移行中」の場合はフレームポートに送信されずにマイグレータ1606の移行プログラム1731に渡される。移行元ディスク装置1603へのアクセスはマイグレータ1606の移行プログラム1731が行う。これらの処理の後、ホスト1601の

I/Oを再開し、マイグレータ1606の移行プログラム1731は移行処理を開始する。

【0108】ポート切替処理終了後、データ移行処理が行われる。最初に、マイグレータ1606の移行プログラム1731は移行元ディスク装置1603上に構成されている論理ボリューム番号と該ボリュームの大きさに従ったボリュームを構築し、使用する変数や、移行ワークテーブル1732の初期化を行う。以後の処理は、移行先ディスク装置の移行プログラムではなく、マイグレータ1606の移行プログラム1731が行うという点を除き、実施例1で説明した処理と同様である。

【0109】データ移行処理が終了した後、実施例3と同様に、スイッチ1602のスイッチ制御プログラム1619は、ポート構成テーブル1318の移行先ディスク装置1304のステータスを「通常」に変更する。オペレータは移行元ディスク装置1603をスイッチ1602から切り離し、移行処理を終了する。

【0110】なお、本実施例ではスイッチと、マイグレータとの間をファイバチャネルで接続したが、ベンダユニークなバス等で接続しても本実施例による効果をえられることは、本発明の技術分野に属する者であれば容易にわかることである。

【0111】〔実施例5〕本実施例では、ホストがスイッチのポート切替を実施する点に特徴がある。本実施例によれば、実施例1乃至実施例4のように、スイッチ上にポート構成テーブルを搭載する必要がない。従って、このような機能を有さないスイッチを用いてSANを構成した場合でも、データ移行を行うことが可能となる。

【0112】図18を用いて、本実施例のコンピュータシステムの構成例を説明する。但し、図1の構成要素と同様のものについては説明を省略する。後述するように、本実施例では、実施例1と異なり、ホスト1801上にポート構成テーブルが搭載され、スイッチ1802にはポート構成テーブルは搭載されない。

【0113】図19は、ホスト1801の構成例を示す。ホスト1801はCPU 1901、メモリ制御部1902、ポート制御部1906、ディスク制御部1904、ドライブ1905、及びメモリ1903から構成され、CPU 1901と、メモリ制御部1902、ポート制御部1906、及びディスク制御部1904との間は内部バス1907で接続されている。CPU 1901は各制御部との間で制御情報、データのやり取りを行う。ポート制御部1906はファイバチャネル1905によりスイッチ1902のポート制御部1812と接続されており、スイッチ1802との間でコマンド、データのやり取りを行う。ホスト1801のポートをNポートと呼べ、スイッチ1902側のポートをFポートと呼ぶ。ドライブ1905はディスク制御部1904と接続されており、CPU 1901からのリード・ライトの指示を受け付ける。ドライブ1905にはホスト1801に必要なプログラムが格納されており、オペレーティングシステム1908、デバイスドライバ1909、アプリケーションプログラム1910及

び、ポート構成テーブル1918が格納される。メモリ1903はメモリ制御部1902と接続されており、CPU 1901からのリード・ライトの要求を受け付ける。オペレーティングシステム1908、ディスク制御部1904、ハードウェアを制御するデバイスドライバ1909、アプリケーション1910等のプログラムは、各プログラムの実行時にディスクドライブ1905からメモリ1903上に読み出される。

【0114】次に、本実施例にデータ移行処理のフロー説明する。そのフローは、図9に示した実施例1のデータ移行処理のフローと同様である。但し、本実施例では、ポート切替処理が実施例1と異なる。ここでは、ポート切替処理のみ説明する。

【0115】本実施例のポート切替処理は、ファブリック401が単体のスイッチで構成されている場合でも複数のスイッチで構成されている場合も同様の動作を行う。最初にオペレータの指示により、ホストのデバイスドライバ1909は、移行元ディスク装置1803を使用している全ホスト1801に対して、ポート切替を開始する旨を通知する。その通知を受け取ったホスト1801のデバイスドライバ1909は移行元ディスク装置1803に対するI/Oをいったん蓄え、ホスト1801のメモリ203上にキューイングする。一方、移行元ディスク装置1803に対して実行中のI/O処理は、終了するまで実行される。移行元ディスク装置1803へのI/Oが停止されると、デバイスドライバ1909は、スイッチ1802に対してI/Oの停止完了を通知する。ポートを切り替える旨を通知したホストのデバイスドライバ1909は、全ホスト1801からのI/O停止完了の通知を受け付けた後、ポート構成テーブル1918の移行元ディスク装置1803の物理ポートID 602と移行先ディスク装置1804の物理ポートID 602を入れ替えるように、全ホストに通知する。以後、ホストのデバイスドライバはフレームの送受信毎にポート構成テーブル1918を参照し、S\_ID 508、D\_ID 507を操作しポートの切替処理を行う。この処理は、実施例1で説明した処理と同様である。

【0116】以上、本発明の実施例1乃至5を説明したが、これらの実施例を適宜組み合わせた形態も考えられる。例えば、実施例3乃至5において、実施例2と同様に移行元ディスク装置1303と移行先ディスク装置1304とを移行用のディスクインターフェースで接続し、これを介してデータ移行を行うことが考えられる。

【0117】

【発明の効果】以上説明したように、本発明によれば、SAN環境に好適なデータ移行方法及びその装置を提供することができる。

【図面の簡単な説明】

【図1】本発明の第1の実施例におけるコンピュータシステムの構成を説明するための図である。

【図2】ホストの構成を説明するための図である。

【図3】一つのスイッチで構成されるファブリックを説明するための図である。

【図4】複数のスイッチで構成されるファブリックを説明するための図である。

【図5】ファイバチャネルのフレームを説明するための図である。

【図6】本発明のポート構成テーブルの構成例を示す。

【図7】本発明の移行元構成情報の構成例を示す。

【図8】本発明の移行ワークテーブルの構成例を示す。

【図9】本発明のデータ移行処理のフローチャートである。

【図10】本発明のポート切替処理のフローチャートである。

【図11】本発明のデータ移行処理のフローチャートである。

【図12】本発明の第2の実施例におけるコンピュータシステムの構成を説明するための図である。

【図13】本発明の第3の実施例におけるコンピュータシステムの構成を説明するための図である。

【図14】本発明の第3の実施例におけるポート構成テーブルの構成例を示す。

【図15】本発明の第3の実施例における移行処理のフローチャートを示す。

【図16】本発明の第4の実施例におけるコンピュータシステムの構成を説明するための図である。

【図17】本発明のマイグレータの構成例を示す。

【図18】本発明の第5の実施例におけるコンピュータシステムの構成を説明するための図である。

【図19】本発明の第5の実施例におけるホストの構成を説明するための図である。

【図20】本発明のポート切替処理前の、各ポートに割り当てられている物理ポートIDと論理ポートIDとの対応を説明するための図である。

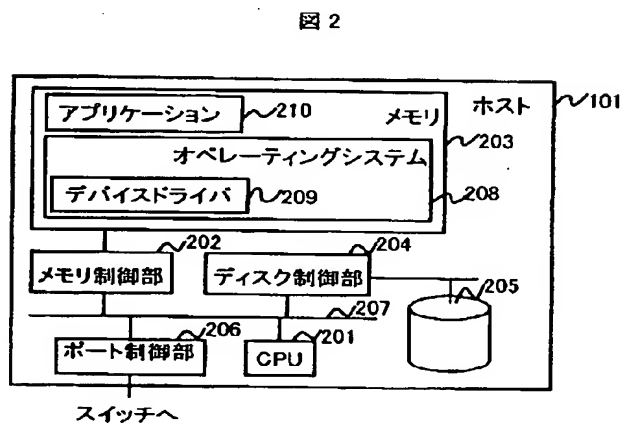
【図21】本発明のポート切替処理後の、各ポートに割り当てられている物理ポートIDと論理ポートIDとの対応を説明するための図である。

【図22】本発明の制御装置をLANに接続した場合の構成例を示す。

【符号の説明】

101, 1201, 1301, 1601, 1801…ホストコンピュータ, 102, 1202, 1302, 1602, 1802…スイッチ, 103, 1203, 1303, 1603, 1803…移行元ディスク装置, 104, 1204, 1304, 1604, 1804…移行先ディスク装置, 105, 1205, 1305, 1605, 1805…ファイバチャネル。

【図2】



【図8】

図8

ボリューム番号	スロット番号	ステータス
1	10	移行中
1	25	移行済み
1	26	移行済み
2	13	移行済み
2	18	移行済み
⋮	⋮	⋮

【図6】

図6

論理ポートID	物理ポートID
0	0
1	1
2	2
3	3
⋮	⋮
N-1	N-1

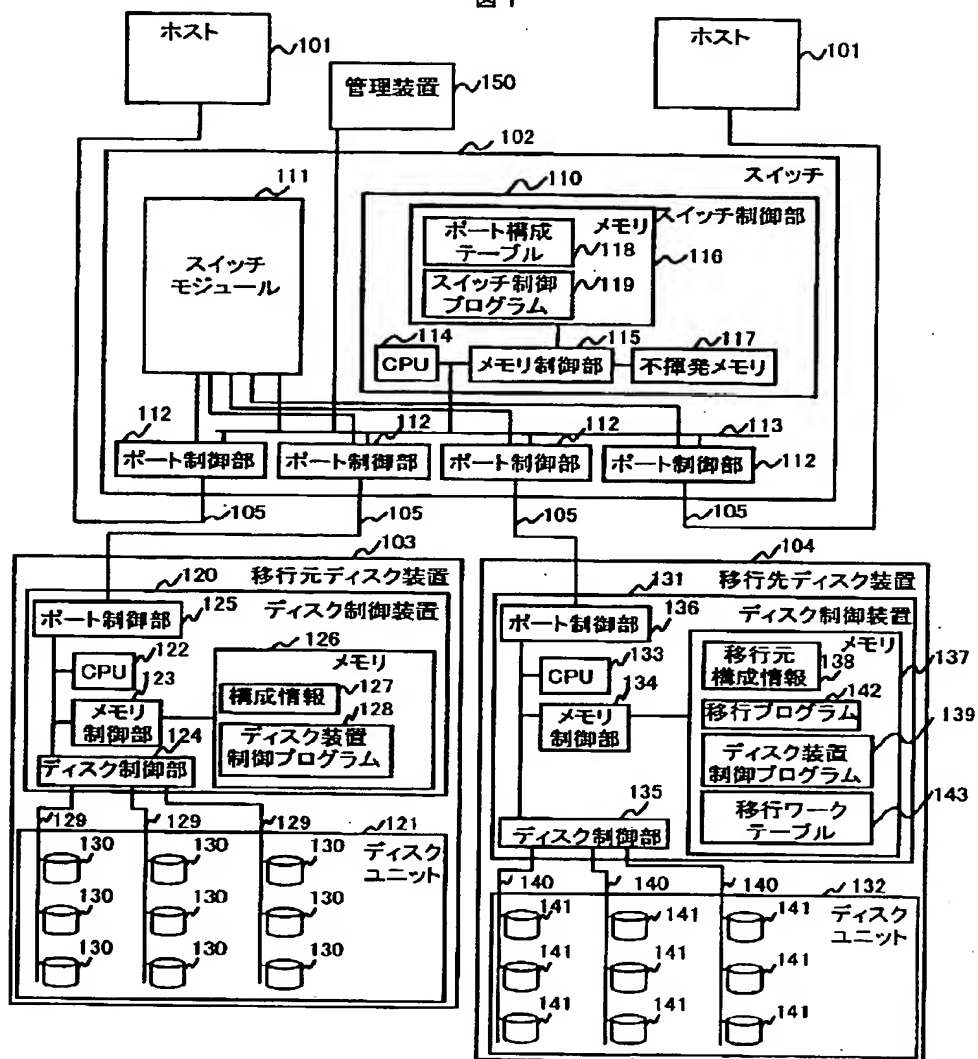
(A) ポート切替処理前

論理ポートID	物理ポートID
0	0
1	2
2	1
3	3
⋮	⋮
N-1	N-1

(B) ポート切替処理後

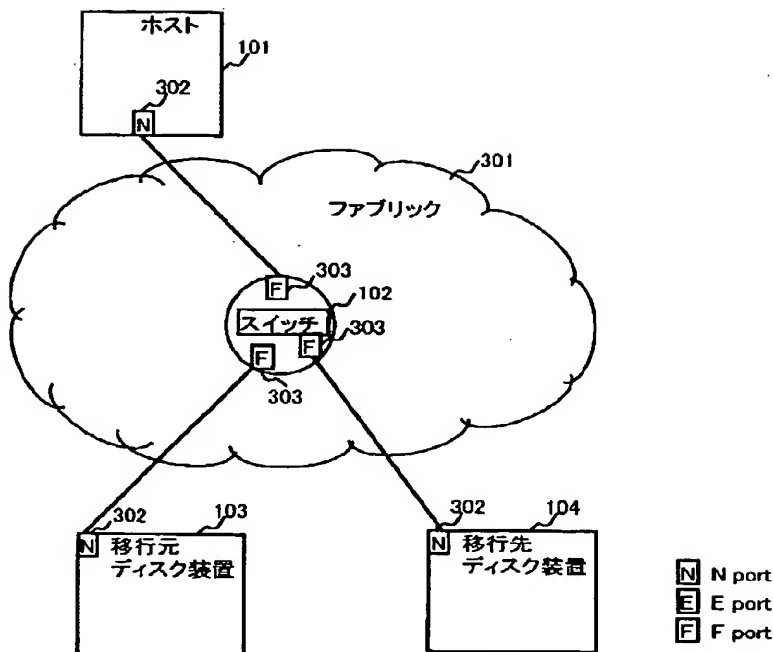
【図1】

図1



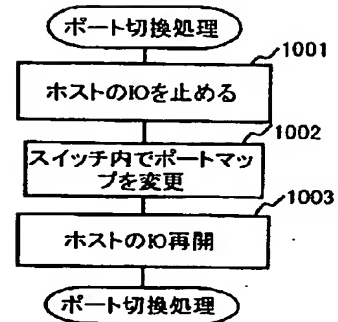
【図3】

図3



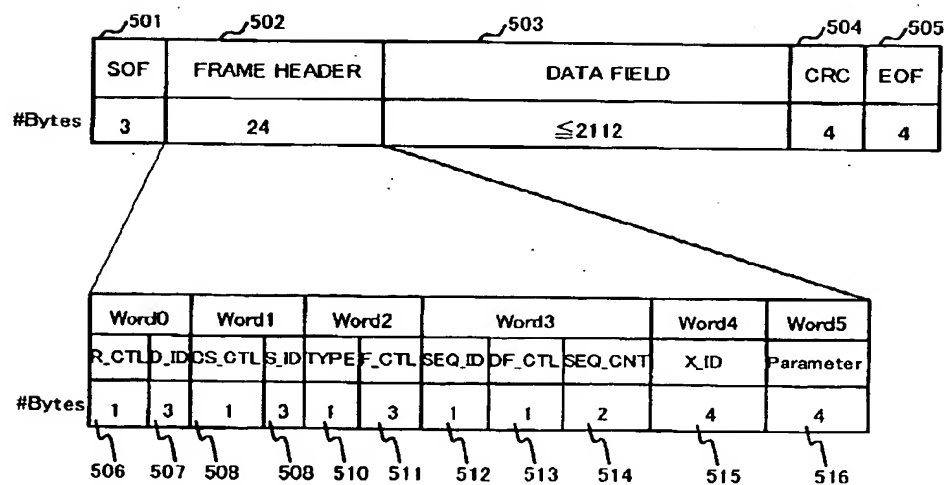
【図10】

図10



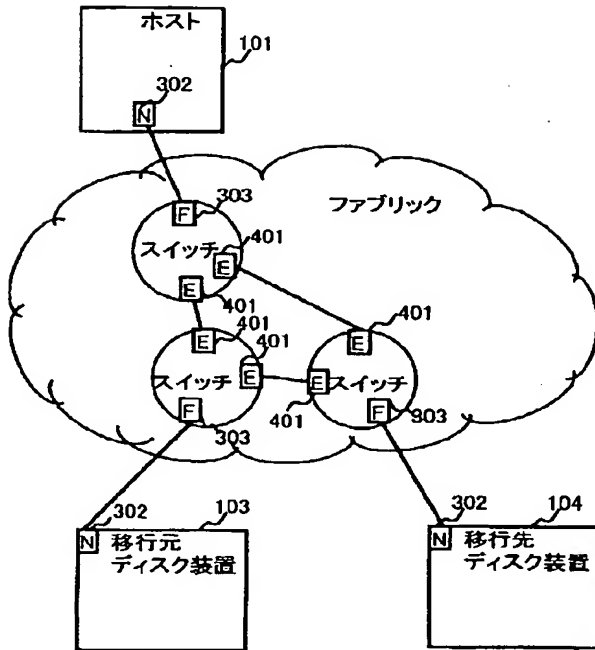
【図5】

図5



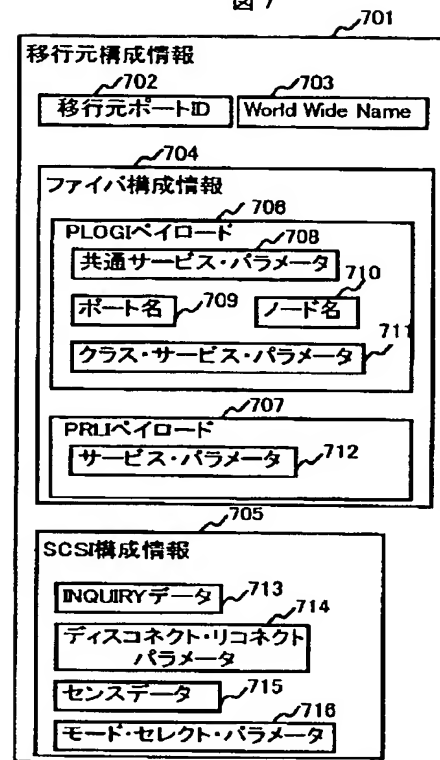
【図4】

図4



【図7】

図7



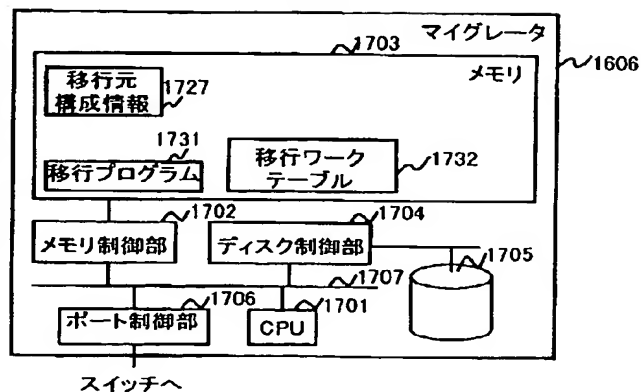
【図14】

図14

論理ポートID	物理ポートID	ステータス
0	0	通常
1	2	移行中
2	1	通常
3	3	通常
⋮	⋮	⋮
N-1	N-1	通常

【図17】

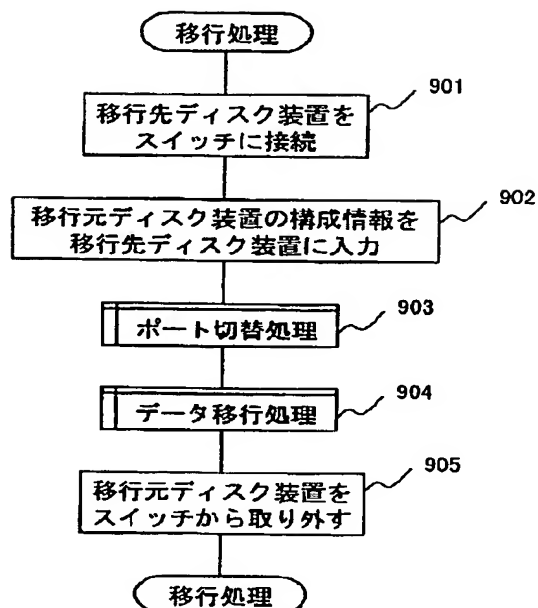
図17





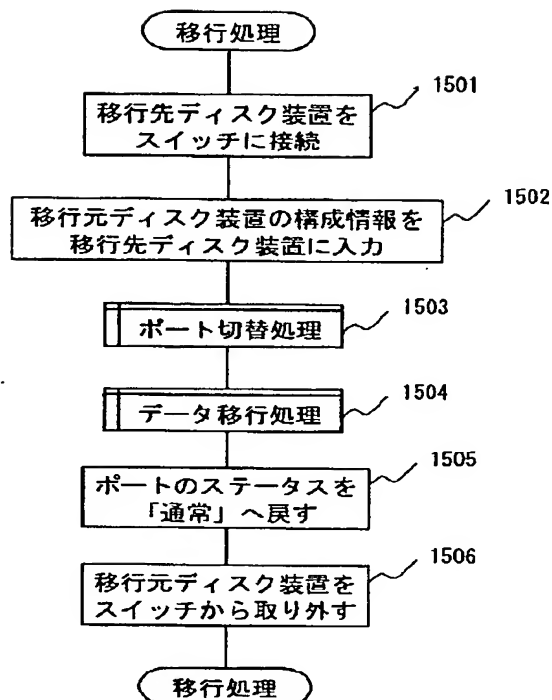
【図 9】

图9



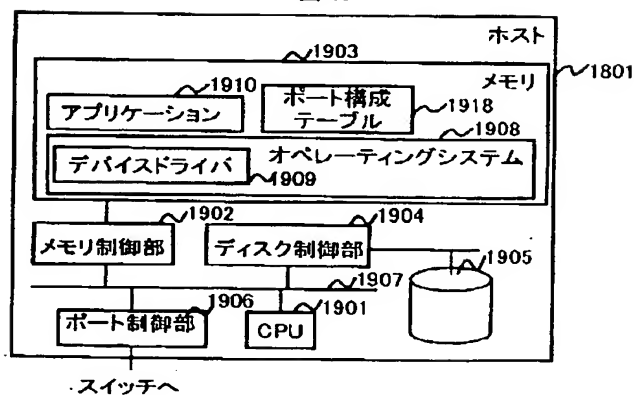
【図 15】

图 15



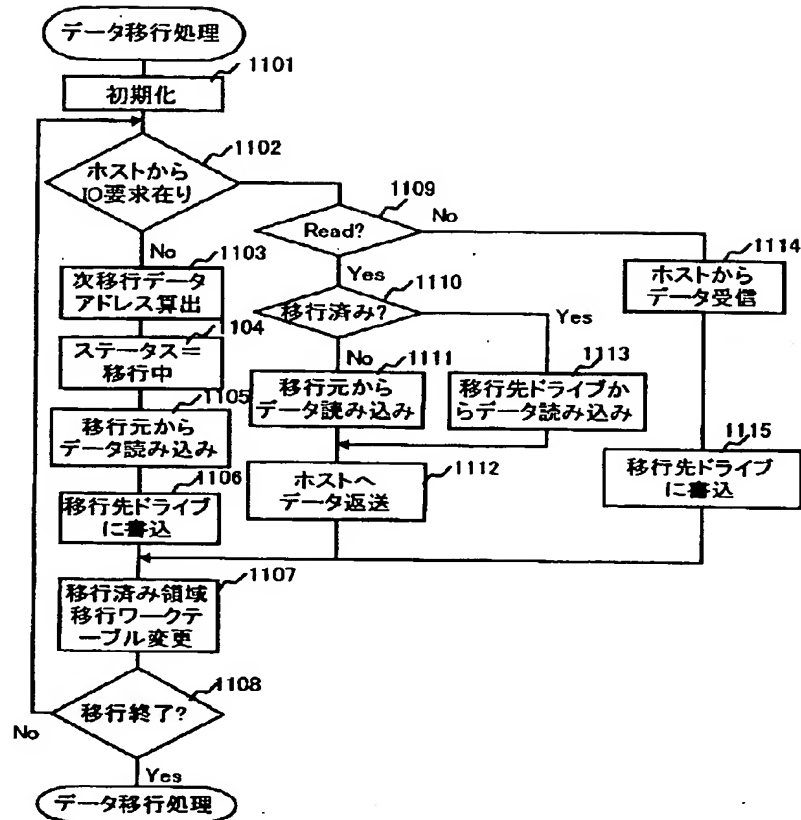
【図 19】

图 19



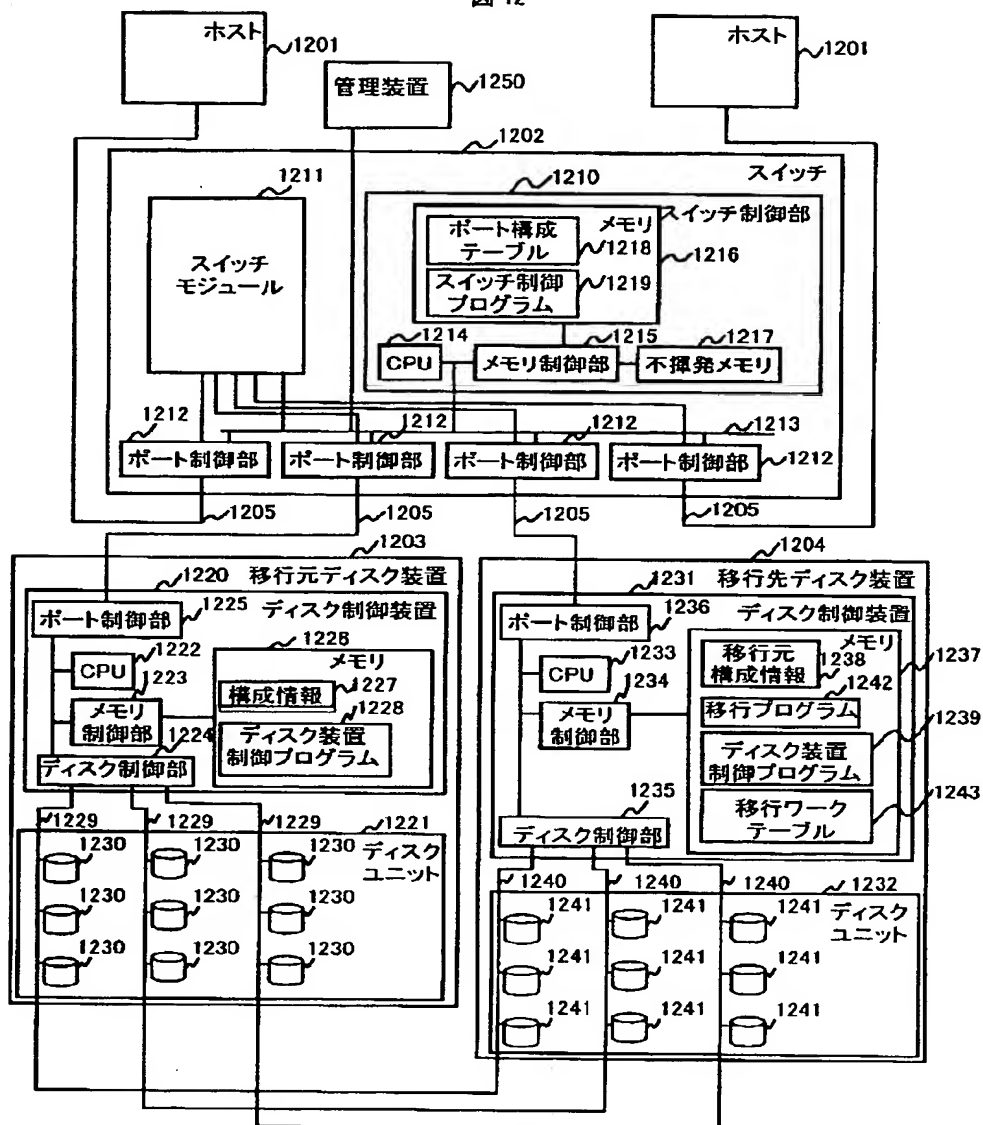
【図11】

図11



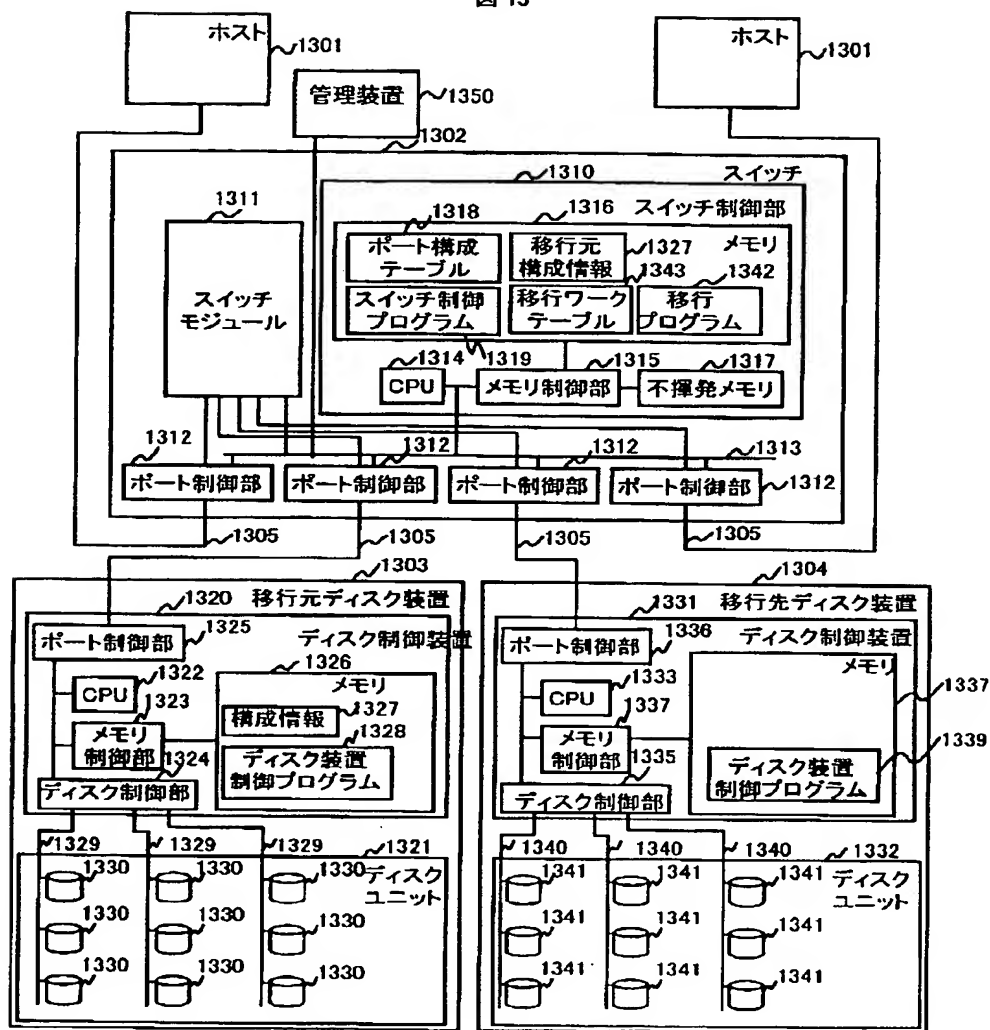
【図12】

図 12



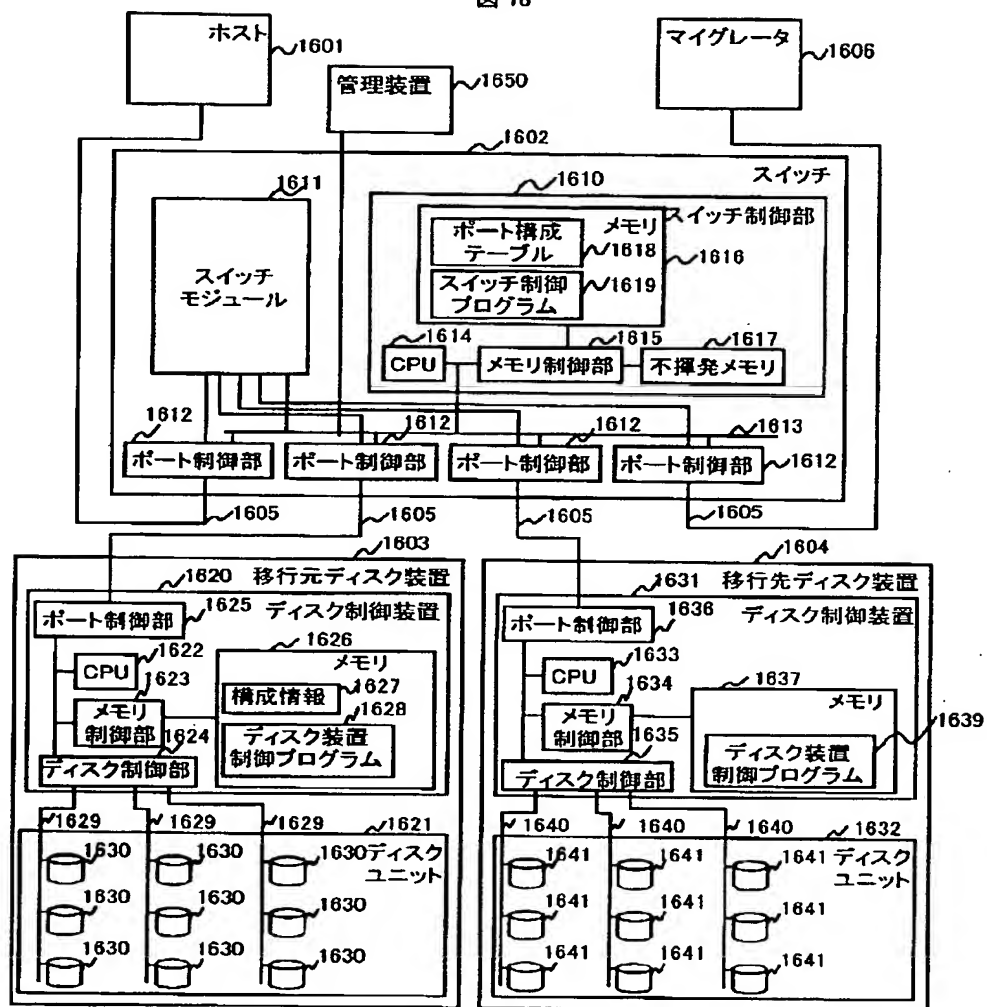
【図13】

図 13



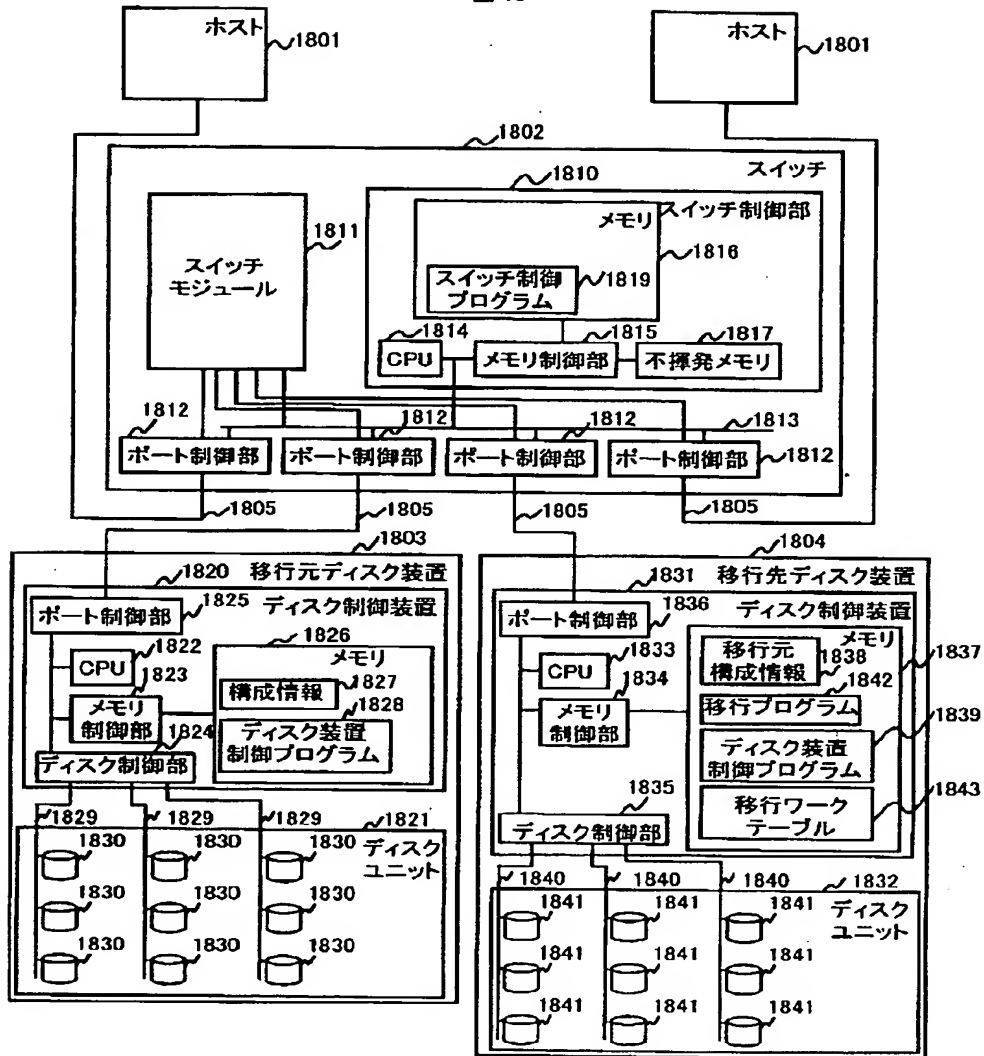
【図 16】

**图 16**



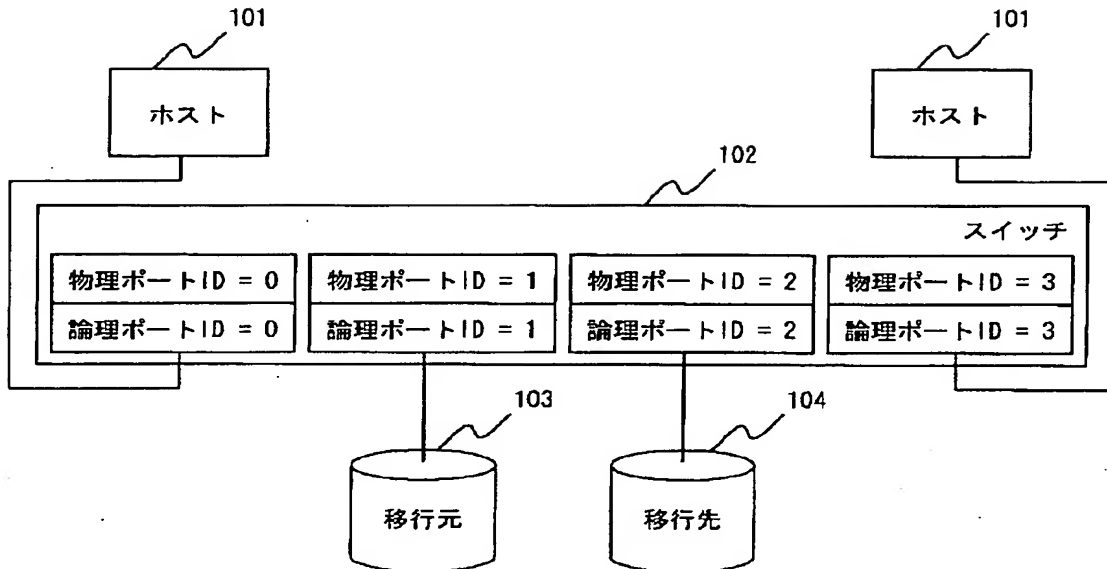
【図18】

図 18



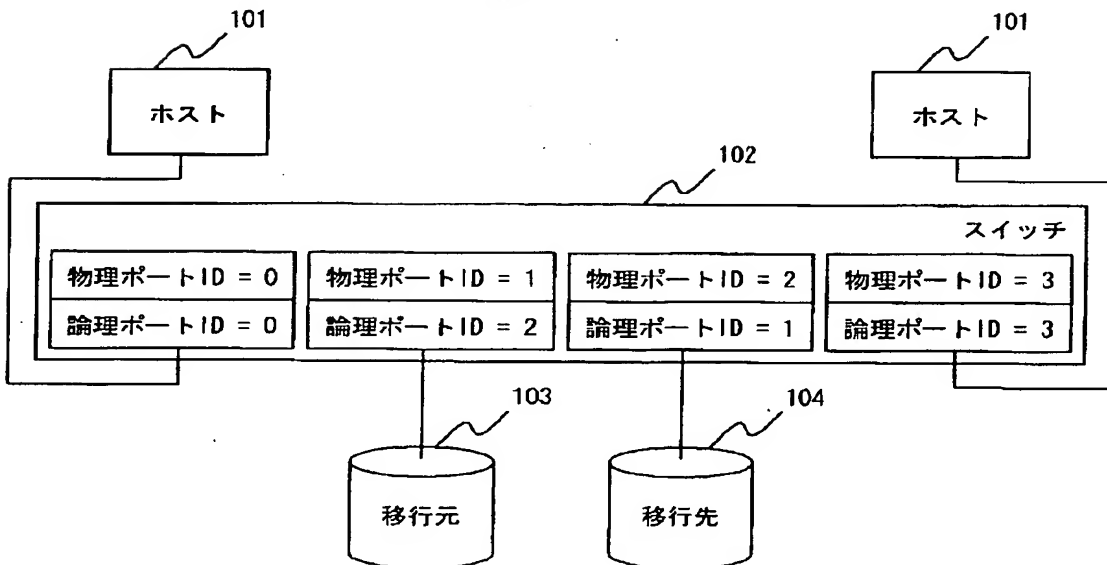
【図20】

図20



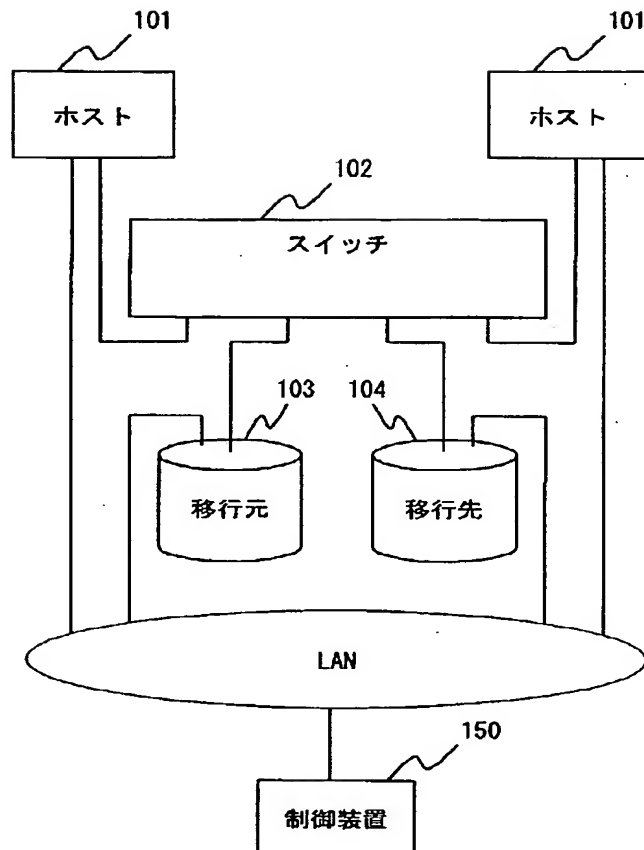
【図21】

図21



【図22】

図22



フロントページの続き

(51)Int.Cl.<sup>7</sup>

G11B 20/10

識別記号

FI

G11B 20/10

テーム(参考)

D

Fターム(参考) 5B014 EA05 EB05 GA25 GC07 GE04  
 HA05 HB27 HC05 HC13  
 5B018 GA04 HA04 MA12 QA20  
 5B065 BA01 CA11 CC03 EA33 PA06 45  
 ZA02 ZA03  
 5B082 CA10  
 5D044 AB01 CC04 DE49 HL01 HL06  
 HL11 JJ06 JJ07



## PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2001-331355

(43)Date of publication of application : 30.11.2001

(51)Int.Cl.

G06F 12/00

G06F 3/06

(21)Application number : 2000-152672

(71)Applicant : HITACHI LTD

(22)Date of filing : 18.05.2000

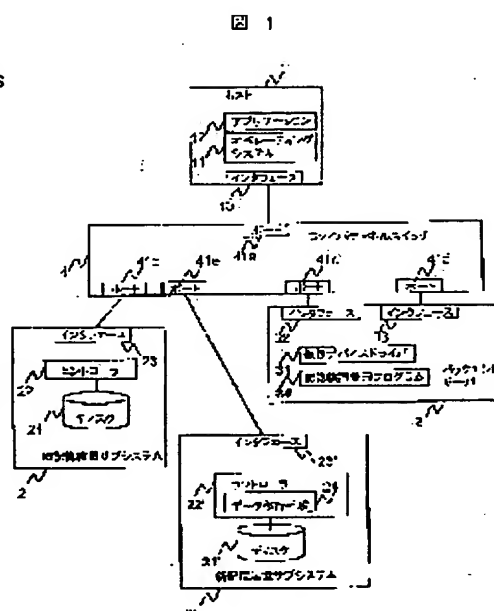
(72)Inventor : KITAMURA MANABU  
ARAI HIROHARU

## (54) COMPUTER SYSTEM

## (57)Abstract:

PROBLEM TO BE SOLVED: To transmissively execute data transition among storage devices to host computers in a computer system, in which plural host computers are connected with plural storage devices.

SOLUTION: A back-end server 3 provides a host 1 having a virtual disk. The virtual disk first appears to be same as an old storage device sub-system 2 to the host 1. When data is transited from the old storage device sub-system 2 to a new storage device sub-system 2', the back-end server 3 first instructs a data transition processing to the new storage device sub-system 2, and after that, switches setting of the virtual disk and makes it correspond to the new storage device sub-system 2'. Data transition among the disk devices is performed transmissively to the host 1, since this switching is transmissively executed to the host 1.



## LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開2001-331355

(P2001-331355A)

(43)公開日 平成13年11月30日(2001.11.30)

(51)Int.Cl. <sup>7</sup>	識別記号	F I	テーマコード*(参考)
G 0 6 F 12/00	5 1 4	G 0 6 F 12/00	5 1 4 5 B 0 6 5
	5 4 5		5 4 5 A 5 B 0 8 2
3/06	3 0 1	3/06	3 0 1 X
	3 0 4		3 0 4 F

審査請求 未請求 請求項の数2 O L (全 7 頁)

(21)出願番号 特願2000-152672(P2000-152672)

(22)出願日 平成12年5月18日(2000.5.18)

(71)出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72)発明者 北村 学

神奈川県川崎市麻生区王禅寺1099番地 株

式会社日立製作所システム開発研究所内

(72)発明者 荒井 弘治

神奈川県小田原市国府津2880番地 株式会

社日立製作所ストレージシステム事業部内

(74)代理人 100075096

弁理士 作田 康夫

Fターム(参考) 5B065 BA01 CE22 EA33 EA35 ZA01

5B082 FA00 HA05

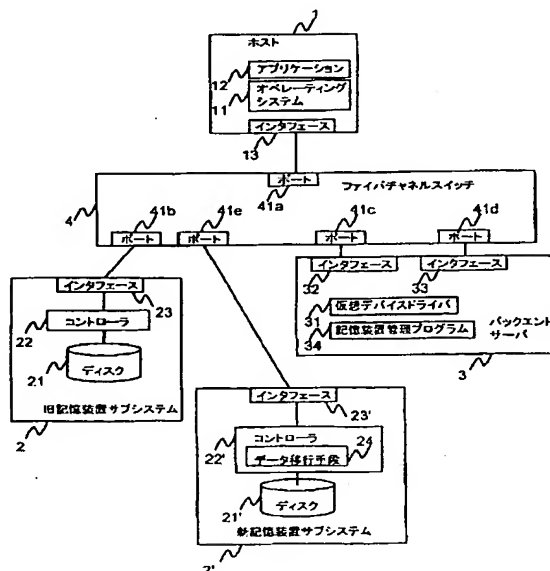
(54)【発明の名称】 計算機システム

(57)【要約】

【課題】複数のホストコンピュータと複数の記憶装置が相互結合された計算機システムにおいて、記憶装置間でのデータの移動をホストコンピュータに対して透過的に実施する。

【解決手段】バックエンドサーバ3はホスト1に対し、仮想ディスクを提供する。仮想ディスクははじめ、旧記憶装置サブシステム2と同じものとしてホスト1に見える。旧記憶装置サブシステム2から新記憶装置サブシステム2'にデータを移行する場合、バックエンドサーバ3は最初新記憶装置サブシステム2'にデータ移行処理を指示し、引き続き仮想ディスクの設定を切り替えて新記憶装置サブシステム2'に対応させる。この切り替えはホスト1に対して透過的に実施されるため、ホスト1に対して透過的にディスク装置間のデータ移行が可能になる。

図 1



【特許請求の範囲】

【請求項1】 複数の計算機と複数の記憶装置と、前記複数の計算機と複数の記憶装置とを相互に結合するスイッチとで構成された計算機システムにおいて、該計算機システムは前記複数の計算機に対し仮想的な記憶装置を提供する手段を有し、前記仮想的な記憶装置は前記複数の記憶装置の少なくとも1つの前記記憶装置に対応する記憶装置であって、前記仮想的な記憶装置を提供する手段は、前記対応を動的に変更することの特徴とする計算機システム。

【請求項2】 請求項1における、仮想的な記憶装置を提供する手段は、前記対応を動的に変更した際に、前記複数の計算機に対しては、前記対応が変化したことを見せないことを特徴とする計算機システム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、情報処理システムなどにおける記憶装置システムのデータアクセス方法に係り、特に、記憶装置内のデータ移行方法に関する。

【0002】

【従来の技術】 パソコン、ワークステーション、メインフレームなどの異なるアーキテクチャ、オペレーティングシステムを採用しているプラットフォームと複数の記憶装置とを相互に接続し、いわゆる1つのネットワークにまとめる動きが盛んになっている。これを一般に、複数の計算機をイーサネット（登録商標）（Ethernet（登録商標））などのネットワークで接続したLAN（Local Area Network）に対する言葉でSAN（Storage Area Network）と呼ぶ。SANは通常ファイバチャネル（Fibre Channel）という光ケーブルないし銅線の伝送路を用いて計算機と記憶装置を接続する。

【0003】 SANにはいくつかの利点があげられている。まず第1に複数の計算機から記憶装置が共通にアクセスできる環境を提供することである。第2に記憶装置同士も相互接続されることにより記憶装置間でのデータ転送が可能で、これにより、ホスト計算機に負荷をかけることなくバックアップや記憶装置間のデータコピーが実現でき、記憶装置障害時には副系の記憶装置への切り替えが可能となる。第3に、これまで個々の計算機に個々の記憶装置が接続されていたため、記憶装置の管理（装置の状態の監視、設定の変更）は接続されている個々の計算機からしかできなかったものを、特定の計算機から全ての記憶装置の管理を可能にする。また、従来のSCSI（Small Computer System Interface）では最高16台までの機器しか接続できなかったが、ファイバチャネルによって100台以上の機器をオンラインで接続でき、容易な拡張性を得られる。

【0004】 近年、SANを実現するための製品が数多く現れてきているが、実際上記利点を生かしたものはない。とくに拡張性においては、機器のオンライン接続は

物理的に可能になったものの、それを活用する基盤技術が不足している。たとえばSANにおいて、ディスク装置の交換のために新規にディスク装置を増設した場合、機器の増設はオンラインにて実施できるが、そのあとでデータの移動をユーザが明示的に行う必要がある。オンラインの機器増設でユーザがメリットを享受するには、単純なハードウェアの増設だけでなく、ハードウェアの増設に伴いデータ移動などがユーザに対して透過的に実施される必要がある。

10 【0005】 ディスク装置間の、オンラインのデータの移動に関しては、米国特許5680640号にその例が開示されている。米国特許5680640号はメインフレーム用ディスクを前提としたデータ移行であるが、ディスク装置間を接続する通信線を利用し、ホストとディスク装置間の接続を短時間切断するだけで、あとはユーザに透過的にディスク装置間のデータ移行を可能にしている。

【0006】

【発明が解決しようとする課題】 米国特許5680640号はユーザに対して限りなく透過的にディスク装置間でのデータ移動を可能にしている。ただし、これはメインフレーム用ディスクを前提としたデータ移行方法であり、SANにおいての適用は出来ない。米国特許5680640号では旧ディスク装置を新規ディスク装置に切り替える際、ディスク装置側の設定によって新規ディスクがあたかも旧ディスク装置であるかのようにホスト側に見せかけることが出来る。これはディスク装置のデバイス番号などの設定を操作することで可能である。

20 【0007】 ただし、SAN、たとえばファイバチャネル環境の場合には、個々のディスクに付与される一意なIDは、ネットワークを構成する機器（ディスク装置、ファイバチャネルスイッチ）同士のネゴシエーションによって決定され、ユーザの設定によって変えられるものではない。米国特許5680640号のデータ移行方法を用いる場合、ホストコンピュータに対して、新規ディスク装置を旧ディスク装置として見せかけることはできず、事実上ユーザに透過的なデータ移行は実現できない。本発明の目的は、ホスト、ユーザに対して透過的で、かつSANの拡張性を生かすことのできるシステムを提供することにある。

40 【0008】

【課題を解決するための手段】 本発明における計算機システムは、ホスト計算機、バックエンド計算機、複数の記憶装置サブシステムと、ホスト計算機とバックエンド計算機とを接続するスイッチとで構成される。ホスト計算機はバックエンド計算機を介して各記憶装置サブシステムにアクセスするが、バックエンド計算機は、ホスト計算機に対して1つないし複数の仮想的なディスク装置を提供する。ホスト計算機から仮想的なディスク装置にアクセス要求があると、バックエンド計算機では要求のあった仮想的なディスク装置の種類に応じて、実際に接

続されている記憶装置サブシステムに適宜要求を出す。

【0009】

【発明の実施の形態】図1は、本発明を適用した計算機システムの一実施形態における構成例を示すブロック図である。計算機システムは、ホスト1、旧記憶装置サブシステム2、新記憶装置サブシステム2、バックエンドサーバ3、ファイバチャネルスイッチ4とで構成される。

【0010】ホスト1はオペレーティングシステム11、アプリケーション12、インタフェース13から構成される。オペレーティングシステム11、アプリケーション12は実際にはホスト1上のCPU、メモリ上で動作するが、これらハードウェアの構成要素については本発明の内容と関係が無いため省略している。実際にはホスト1以外に複数のホストコンピュータがつながる環境が一般的であるが、本発明では簡単のため、ホストコンピュータとしてホスト1のみを記載している。旧記憶装置サブシステム2はディスク21、コントローラ22、インタフェース23とから構成される。ディスク21は複数の物理ディスクをまとめて1つの論理的なディスク装置に見せかけた論理ドライブであっても、本発明の内容に変わりはない。インタフェース23はファイバチャネルスイッチ4と接続される。新記憶装置サブシステム2も旧記憶装置サブシステム2と同様、ディスク21、コントローラ22、インタフェース23とから構成される。旧記憶装置サブシステム2と異なる点は、コントローラ22中にデータ移行手段24が含まれることである。

【0011】バックエンドサーバ3は仮想デバイスドライバ31、インタフェース32、33から構成される。仮想デバイスドライバ31はバックエンドサーバ3上のCPU、メモリ上で動作するソフトウェアで、ユーザによって外部から設定を変更したりあるいはプログラム自体の入れ替えをすることが可能であるが、CPU、メモリなどのハードウェア構成要素に関しては本発明の内容と関係ないため省略している。

【0012】ファイバチャネルスイッチ4は複数のポート41a、41b、41c、41d、41e（以下総称してポート41と略す）から構成され、ホスト1、旧記憶装置サブシステム2、新記憶装置サブシステム2、バックエンドサーバ3を相互に接続するために使用される。ポート41aからはいずれもポート41b、41c、41d、41eにアクセスすることが可能である。そのため、ホスト1はポート41b、41eから直接旧記憶装置サブシステム2や新記憶装置サブシステム2にアクセスすることもできるが、本実施形態においては、基本的にホスト1はすべてバックエンドサーバ3を介して記憶装置サブシステム2にアクセスすることとする。

【0013】バックエンドサーバ3の役割について説明する。バックエンドサーバ3は仮想デバイスドライバ3

1によって、ホスト1から見てあたかも1つないし複数のディスク装置であるかのように見える。本実施形態では、ホスト1がポート41dを介してインタフェース33を見ると、1つのディスクがつながっているように見えるものとする。以降、このディスクのことを仮想ディスクと呼ぶ。仮想デバイスドライバ31は、最初はホスト1からは仮想ディスクが旧記憶装置サブシステム2のディスク21と同じ物に見えるように設定されている。すなわちホスト1が仮想ディスクの論理ブロックアドレス(LBA)0あるいはLBA1にアクセスすると、仮想デバイスドライバ31はインタフェース32、ポート41cを介して、ディスク21のLBA0あるいはLBA1にアクセスし、結果をインタフェース33、ポート41dを介してホスト1に返す。本発明の実施形態ではインタフェース32がディスク21やディスク21にアクセスするために使われ、またインタフェース33がホスト1とのやり取りに使われるようになっているが、1つのインタフェースでこれら2つの役割を行わせることも可能である。また、仮想デバイスドライバ31の設定を変えることで、仮想ディスクが新記憶装置サブシステム2のディスク21に見えるようにすることも可能である。設定変更を行った場合、ホストコンピュータから見える仮想ディスクに変化はない。ファイバチャネルインタフェースを持つディスク装置の場合、ホストコンピュータからはポートIDと論理ユニット番号(LUN)で一意にディスク装置が認識できるが、仮想デバイスドライバの設定を変更して仮想ディスクがディスク21からディスク21に変更されたとしても、ホスト1に対して見える仮想ディスクのポートIDとLUNは変化せず、ホスト1は実際にアクセスしているディスクが変わったことの認識はない。次に新記憶装置サブシステム2のデータ移行手段24について説明する。データ移行手段24は米国特許5680640号に開示されているものと同様の手段を有する。データの移行が指示されると、データ移行手段24は記憶装置サブシステム2のディスク21の先頭から順にデータを読み出し、ディスク21へとデータを書き込む。さらに各ブロックないしは複数ブロック単位に、データの移行が終了したかどうかを記録するテーブルを持ち、移行処理中にリードアクセスがくると、このテーブルを参照し、データ移行が済んでいない領域についてはディスク21からデータを読み出し、データ移行が済んでいる領域についてはディスク21のデータを返す。

【0014】図2はデータ移行手段のもつテーブル100を表したものである。データ移行手段24ではブロックごとにデータをディスク21からディスク21へとコピーしていく。テーブル100ではそれぞれのアドレス101ごとにフラグ102を持つ。フラグ102が1である場合には、そのアドレスのデータはすでにディスク21からディスク21にコピーされたことを示し、0の場合には未コピーであることを示す。データ移行処理

や、データ移行処理中のリード、ライト処理ではこのテーブル100を利用する。

【0015】図3で、データ移行手段24の行う移行処理の流れを説明する。まずカウンタBを用意し、初期値を0とする(ステップ2001)。次にテーブル100を参照し、LBA Bのフラグ102が1かどうかチェックする(ステップ2002)。フラグが1の場合にはデータ移行が済んでいるため、カウンタBを1増加する(ステップ2005)。また、ステップ2002で、フラグ102が0であれば、ディスク21からディスク21へとデータをコピーし(ステップ2003)、テーブル100の該当するフラグ102を1に更新し(ステップ2004)、ステップ2005へと進む。ステップ2006ではディスク21の最終LBAまで処理したかチェックする。すなわちBがディスク21の最終LBAを超えたかどうかチェックし、超えていれば処理を完了し、超えていなければステップ2002に戻って、処理を繰り返す。

【0016】次に図4で、データ移行手段が図3のデータ移行処理を行っている間に、上位ホスト、すなわち本実施形態ではバックエンドサーバ3からのライト要求があった場合の処理を説明する。この処理は簡単で、ステップ2101でディスク21にデータを書き込み、ステップ2102でテーブル100の該当するLBAのフラグ102を1に更新する。つまり、データ移行処理中にライト処理が行われたLBAについては、ディスク21からのデータ移行は行われない。

【0017】図5で、データ移行手段が図3のデータ移行処理を行っている間に、上位ホスト、すなわち本実施形態ではバックエンドサーバ3からのリード要求があった場合の処理を説明する。ステップ2201で、テーブル100内のリード要求のあったLBAについてフラグ102を参照する。ステップ2202でフラグ102が1かどうかチェックして処理を分岐する。フラグが1の場合には、そのLBAについてはディスク21からのデータ移行が完了しているの、ディスク21からデータを読み出す(ステップ2203)。フラグが0の場合には、そのLBAについてデータ移行が完了していないので、一旦ディスク21からディスク21にデータをコピーする(ステップ2205)。続いてテーブル100のフラグ102を1に更新して(ステップ2206)、ステップ2203以降へ進む。ステップ2204で読み出したデータをバックエンドサーバ3に渡して処理は完了する。

【0018】次に、本実施形態のシステムでの、旧記憶装置サブシステム2から新記憶装置サブシステム2へのデータ移行処理について、システム全体の流れを説明していく。データ移行を行う際、ユーザはバックエンドサーバ3に移行を指示する。バックエンドサーバ3から新記憶装置サブシステム2へのデータ移行処理開始の指示は、インタフェース32を介して新記憶装置サブシステム

2に伝えられる。図6はバックエンドサーバ3の処理の流れを説明している。バックエンドサーバ3は移行の指示を受けると、まず、仮想デバイスドライバ31による仮想ディスクの動作を停止する(ステップ100

1)。これにより、仮想デバイスドライバ31から旧記憶装置サブシステム2へのアクセスは中断され、仮想デバイスドライバ31はホスト1から仮想ディスクに対するアクセスコマンドを受け付けても、アクセス中止が解除されるまで応答を返さない。次に記憶装置管理プログラム34は新記憶装置サブシステム2に対してデータ移行処理の開始を指示する(ステップ1002)。新記憶装置サブシステム2の行うデータ移行処理については後述する。ステップ1003では、仮想デバイスドライバ31がこれまでホスト1に見せていた仮想デバイスの設定を、ディスク21へのアクセスを行うように変更し、ステップ1004ではステップ1001で中止していたアクセスを再開させる。仮想ディスクのアクセスが再開されると、ステップ1001、ステップ1002の間にホスト1から仮想ディスクに対してアクセスがきていた場合、そのアクセスは全てディスク21に対して実施される。

【0019】また、本実施形態においては、記憶装置サブシステム2とバックエンドサーバ3が直接スイッチにつながった接続形態であったが、図7のように記憶装置サブシステム2がバックエンドサーバ3を介してつながる構成であっても実現は可能である。さらに、新規に増設する記憶装置サブシステム2が図1の例のようにデータ移行手段24をもたないような場合には、図8のようにバックエンドサーバ3側にデータ移行手段24を持たせ、バックエンドサーバでデータ移行処理を行わせることで同様のことが実現できる。また、本実施形態ではバックエンドサーバ3を設けてホストに仮想的なディスクを見せる処理を施しているが、図9のように仮想デバイスドライバ31、記憶装置管理プログラム34、そしてデータ移行手段24をファイバチャネルスイッチ4に持たせるという構成も可能である。本実施形態では、ホストコンピュータに透過的にディスク間のデータ移行ができる例を示したが、さまざまな適用先がある。仮想デバイスドライバが仮想ディスクと実際の記憶装置との対応付けを行えば、ホストコンピュータにとっては実際にデータがどの記憶装置にあってもかまわない。そのため、例えば普段は必要最低限の仮想ディスクを定義しておき、必要になった時点で動的に必要な容量の仮想ディスクを用意できるような記憶装置管理システムや、データのアクセス頻度により、ホストに透過的にデータを低速ディスクから動的に高速ディスクに移動するシステムなどに応用できる。

【0020】

【発明の効果】本発明によれば、ホストコンピュータに対して一切透過的にディスク装置間のデータ移動や、デ

ディスク容量のオンライン拡張など、あらゆるデータ操作が可能となる。

【図面の簡単な説明】

【図 1】 本発明の実施形態における計算機システムの構成例を示すブロック図である。

【図 2】 本発明の新記憶装置サブシステムのデータ移行手段の使用するテーブルを示すテーブル構成図である。

【図 3】 本発明のデータ移行手段が行うデータ移行処理の流れを示すフローチャートである。

【図 4】 データ移行処理中にライト要求が来たときの、データ移行手段の処理の流れを示すフローチャートである。

【図 5】 データ移行処理中にリード要求が来たときの、データ移行手段の処理の流れを示すフローチャートである。

【図 6】 本発明の実施形態における計算機システムにおいて、旧記憶装置サブシステムから新記憶装置サブシステムへのデータ移行処理を行うときの、バックエンドサ

ーバの処理の流れを示すフローチャートである。

【図 7】 本発明の実施形態を実現する、別の計算機システムの構成例を示すブロック図である。

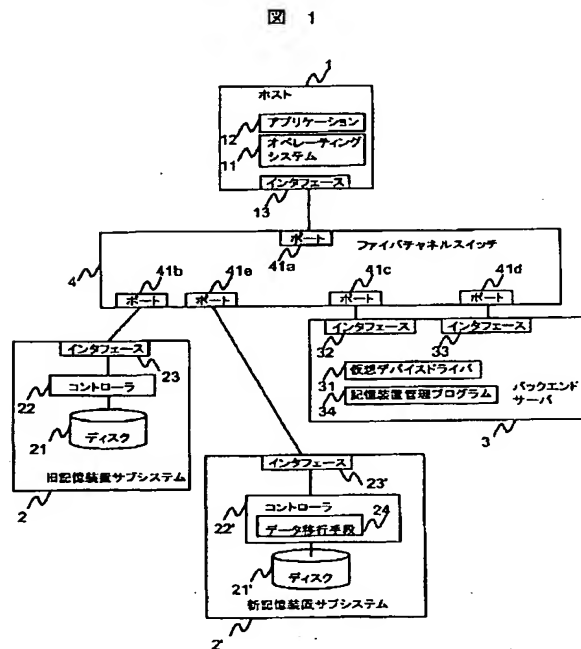
【図 8】 本発明の実施形態を実現する、別の計算機システムの構成例を示すブロック図である。

【図 9】 本発明の実施形態を実現する、別の計算機システムの構成例を示すブロック図である。

【符号の説明】

- 1…ホスト、2…旧記憶装置サブシステム、2'…新記憶装置サブシステム、3…バックエンドサーバ、4…ファイバチャネルスイッチ、11…オペレーティングシステム、12…アプリケーション、13…インタフェース、21…ディスク、22…コントローラ、23…インタフェース、24…データ移行手段、31…仮想デバイスドライバ、32…インタフェース、33…インタフェース、34…記憶装置管理プログラム、41a…ポート、41b…ポート、41c…ポート、41d…ポート、41e…ポート。

【図 1】

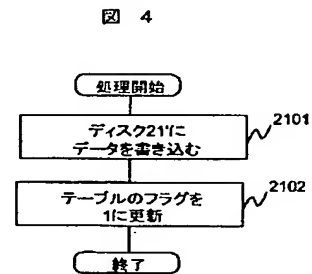


【図 2】

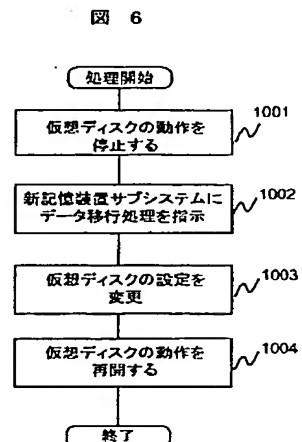
図 2

アドレス	フラグ
LBA0	1
LBA1	1
LBA2	0
LBA3	0
LBA4	1
...	...
LBA <sub>n</sub>	0

【図 4】

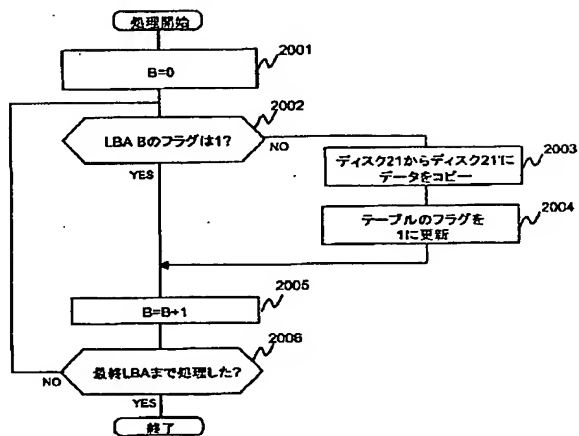


【図 6】



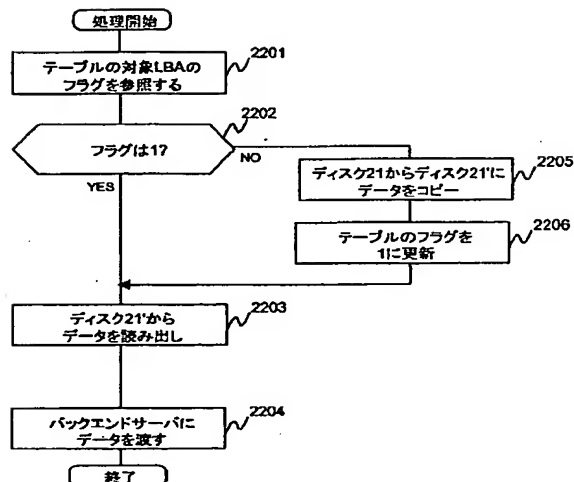
【図3】

図 3



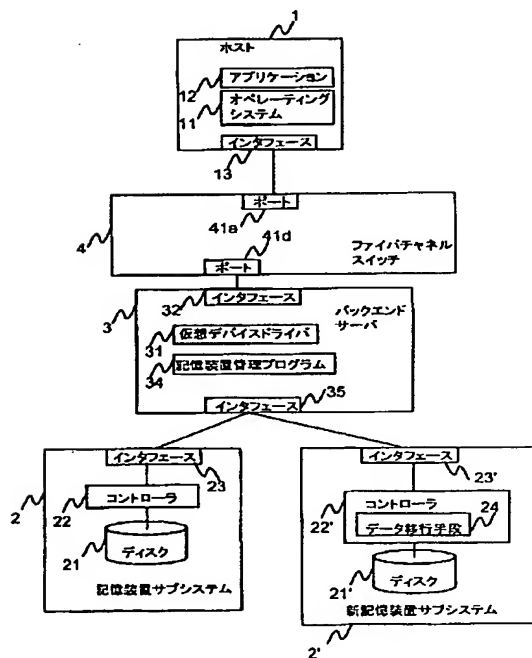
【図5】

図 5



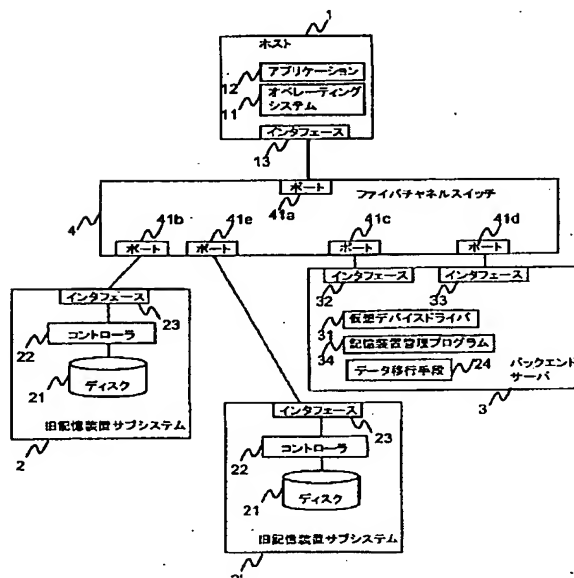
【図7】

図 7



【図8】

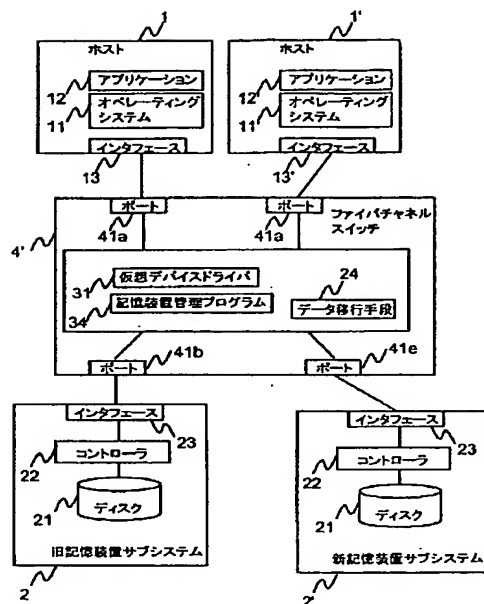
図 8





【図9】

図 9



US09/991219

COMPUTER SYSTEM

BACKGROUND OF THE INVENTION

The present invention relates to a data access method to a storage apparatus system in an information processing system or the like. More particularly, it relates to a data migration method within the storage apparatus system.

An increasingly prevailing trend is that a plurality of platforms and a plurality of storage apparatuses are connected to each other so that the platforms and the storage apparatuses will be integrated into so-called a single network. Here, the platforms are, for example, a personal computer, a workstation, and a mainframe which employ mutually different architectures and operating systems. This resultant network is generally referred to as "SAN (Storage Area Network)", which is a technical term corresponding to "LAN (Local Area Network)" formed by connecting a plurality of computers by a network such as Ethernet. In SAN, usually, the computers and the storage apparatuses are connected using a communication line referred to as "Fibre Channel" formed of an optical cable or a copper wire.

It has been pointed out that SAN has several advantages. The first advantage is to provide an environment where the storage apparatuses can be

accessed from the plurality of computers in common. The second advantage is that the data transfers among the storage apparatuses are made possible by inter-connecting the storage apparatuses among themselves.

5 This makes it possible to implement backups or data copies among the storage apparatuses without imposing a load onto the host computers, thereby, at the time of the storage apparatuses' failure, allowing the switching to storage apparatuses included in a secondary system. Thirdly, until now, the individual storage apparatuses have been connected to the individual computers, respectively. As a result, managing the individual storage apparatuses (i.e., monitoring the states of the apparatuses, and modifying the settings thereof) has been possible only from the individual computers connected thereto. In SAN, however, it becomes possible to manage all the storage apparatuses from an arbitrarily specified computer. Also, the conventional SCSI (Small Computer System Interface) could connect only 16 appliances at the maximum.

20 Meanwhile, Fibre Channel can connect 100 or more appliances on-line, thereby making it possible to obtain an easy scalability.

In recent years, a large number of products for implementing SAN have appeared. There exists, however, none of the products that actually make full utilization of the above-described advantages. In the scalability in particular, although the on-line connec-

tion of the appliances has been made physically possible, fundamental technologies for taking full advantage of the scalability are lacking. In SAN, for example, when additionally installing a new disk apparatus for the replacement of an existing disk apparatus, the additional installation of the appliance can be carried out on-line. Concerning the data migration, however, the users usually need to explicitly instruct the data migration into the new disk apparatus after switching the connection off-line once. For the users to enjoy the merits by the appliance additional-installation during the on-line, in addition to the mere hardware additional-installation, it is also required to carry out the data migration or the like transparently to the users, i.e., in such a manner that the users are unconscious of the data migration or the like, in accompaniment with the hardware additional-installation.

Concerning the on-line data migration among the disk apparatuses, U.S. Patent No. 5,680,640 has disclosed its example. The example in U.S. Patent No. 5,680,640, which describes the data migration where the disk apparatuses designed for a mainframe are assumed, utilizes communication lines for connecting the disk apparatuses, and only disconnects the connections among the hosts and the disk apparatuses for a short time. After that, the example allows the data migration among the disk apparatuses to be implemented transparently to

the users.

U.S. Patent No. 5,680,640 allows the data migration among the disk apparatuses to be implemented unlimitedly transparently to the users. However, this is the data migration method where the for-mainframe-designed disk apparatuses are assumed, and thus this method is inapplicable to SAN. In the for-mainframe-designed disk apparatuses as disclosed in U.S. Patent No. 5,680,640, when replacing an old disk apparatus by a new disk apparatus, the setting on the disk apparatus side makes it possible to make the new disk apparatus look as if it were the old disk apparatus when seen from the host side. This is made possible by manipulating the setting of device numbers or the like of the disk apparatuses.

However, in SAN, especially in the case of, e.g., the Fibre Channel environment, the individual unique IDs assigned to the individual disk apparatuses are determined by the negotiation among the appliances (i.e.; the disk apparatuses, and a fibre channel switch) included in the network. Accordingly, the setting made by the users never changes the IDs. This condition, when using the data migration method in U.S. Patent No. 5,680,640, makes it impossible to make the new disk apparatus disguise the old disk apparatus with respect to the host computers. Consequently, in reality, it is impossible to implement the data migration that is transparent to the hosts and the users.

#### SUMMARY OF THE INVENTION

It is an object of the present invention to provide a system that is transparent to the hosts and the users, and that is capable of making full utilization of the scalability of SAN.

The computer system in the present invention includes, for example, host computers, a back end computer (back end server), a plurality of storage subsystems, and a switch for connecting at least the host computers with the back end computer. The host computers access each storage subsystem via the back end computer. Here, the back end computer provides one virtual disk apparatus or a plurality of virtual disk apparatuses to the host computers. If the host computers issue access requests to the virtual disk apparatus/apparatuses, the back end computer issues an appropriate request to the storage subsystems connected thereto actually, depending on the type of the virtual disk apparatus/apparatuses to which the requests have been issued.

According to the present invention, it becomes possible to implement all the data manipulations, such as the data migration among the disk apparatuses and on-line extension of the disk capacities, completely transparently to the host computers.

#### BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a block diagram for illustrating the configuration example of a computer system in an embodiment of the present invention;

5 FIG. 2 is a table configuration diagram for illustrating a table used by a data migration unit toward a new storage subsystem of the present invention;

FIG. 3 is a flow chart for illustrating the flow of a data migration processing performed by the data migration unit of the present invention;

FIG. 4 is a flow chart for illustrating the flow of a processing performed by the data migration unit when a writing request arrives during the data migration processing;

FIG. 5 is a flow chart for illustrating the flow of a processing performed by the data migration unit when a reading request arrives during the data migration processing;

20 FIG. 6 is a flow chart for illustrating the flow of a processing performed by a back end server when the data migration processing from an old storage subsystem to the new storage subsystem is performed in the computer system in the embodiment of the present invention;

25 FIG. 7 is a block diagram for illustrating the configuration example of another computer system embodying another embodiment of the present invention;

FIG. 8 is a block diagram for illustrating the configuration example of another computer system embodying another embodiment of the present invention; and

5 FIG. 9 is a block diagram for illustrating the configuration example of another computer system embodying another embodiment of the present invention.

#### DESCRIPTION OF THE EMBODIMENTS

FIG. 1 is a block diagram for illustrating the configuration example in an embodiment of a computer system to which the present invention is applied. The computer system includes hosts 1, 1', an old storage subsystem 2, a new storage subsystem 2', a back end server 3, and a fibre channel switch 4.

15 The host 1 includes an operating system 11, an application program 12, and an interface 13. Although, actually, the operating system 11 and the application program 12 operate on a CPU 14 and a memory 15 on the host 1, the detailed explanation will be omitted concerning these hardware configuration components. Actually, the general environment is a one where, like the host 1', a plurality of host computers other than the host 1 are connected, but only the operation of the host 1 as the host computers will be described for simplifying the operation of the present invention. In the host 1', the components with apostrophe are the same as or equivalent to the

corresponding components in the host 1.

The old storage subsystem 2 includes a disk 21, a controller 22, and an interface 23. Even if the disk 21 is a logical drive that is made to disguise a single logical disk apparatus by integrating a plurality of physical disks, the contents of the present invention remain unchanged. The interface 23 is connected to the fibre channel switch 4.

As is the case with the old storage subsystem 2, the new storage subsystem 2' also includes a disk 21', a controller 22', and an interface 23'. The point that differs from the old storage subsystem 2 is that a data migration unit 24 is included in the controller 22'.

The back end server 3 includes a virtual device driver 31 and interfaces 32 and 33. The virtual device driver 31 is a software that operates on a CPU 35 and a memory 36 on the back end server 3, and it is possible for the user to modify the setting thereof from the outside or to replace the program itself. The detailed explanation will be omitted regarding the hardware configuration components such as the CPU 35 and the memory 36.

The fibre channel switch 4, which includes a plurality of ports 41a, 41a', 41b, 41c, 41d, and 41e (hereinafter, abbreviated generically as a port 41), is used for interconnecting the host 1, the old storage subsystem 2, the new storage subsystem 2', and the back

end server 3. Accessing any of the ports 41a', 41b, 41c, 41d, and 41e is possible from the port 41a. This condition allows the host 1 to directly access the old storage subsystem 2 and the new storage subsystem 2' from the ports 41b and 41e, respectively. In the present embodiment, however, it is assumed that, basically, the host 1 accesses the old storage subsystem 2 and the new storage subsystem 2' all through the back end server 3.

The explanation will be given below concerning the role of the back end server 3. The virtual device driver 31 makes the back end server 3 look as if it were one disk apparatus or a plurality of disk apparatuses when seen from the host 1. In the present embodiment, it is assumed that, when the host 1 looks at the back end server 3 through the port 41d, the host 1 sees one disk apparatus as if it were connected to the port 41d. Hereinafter, this disk apparatus will be referred to as a virtual disk.

The virtual device driver 31 is set so that, when seen from the host 1, the virtual disk at first looks the same as the disk apparatus 21 in the old storage subsystem 2. Namely, if the host 1 accesses, for example, a logical block address (LBA) 0 in the virtual disk, the virtual device driver 31 accesses LBA 0 in the disk apparatus 21 via the interface 32 and the port 41c, then returning the result back to the host 1 via the interface 33 and the port 41d. In the embodi-

ment of the present invention, although the interface 32 is used for accessing the disk 21 or the disk 21' and also the interface 33 is used for the communications with the host 1, it is possible to cause a single interface to play these two roles.

Here, attention should be paid to the following point: Modifying the setting of the virtual device driver 31 makes it possible to make the virtual disk look as if it were the disk apparatus 21' in the new storage subsystem 2'. When this setting modification is performed, the virtual disk that the host computer sees remains unchanged. Even if the virtual disk is modified from the disk 21 to the disk 21' by the setting modification of the virtual device driver 31, the port ID and LUN of the virtual disk that the host 1 is made to see remain unchanged. As a result, the host 1 has none of the recognition that the actually accessed disk has been changed.

Next, the explanation will be given below regarding the data migration unit 24 in the new storage subsystem 2'. The data migration unit 24 has a unit similar to the unit disclosed in U.S. Patent No. 5,680,640. If the data migration is instructed, the data migration unit 24 reads the data sequentially from the front-head of the disk 21 in the old storage subsystem 2, then writing the data into the disk 21' in the new storage subsystem 2'. Moreover, the unit 24 has a table (i.e., 100 in FIG. 2) for recording whether

or not the data migration has been finished in an each-block unit or in a plural-block unit. If a reading access arrives from the host computer during the data migration processing, the unit 24 makes reference to this table. Then, the unit 24 reads the data from the disk 21 in the old storage subsystem 2 with respect to regions where the data migration is not over, and returns the data in the disk 21' in the new storage subsystem 2' with respect to regions where the data migration is over.

FIG. 2 illustrates the table that the data migration unit 24 has. The data migration unit 24 copies the data on each block basis from the disk 21 in the old storage subsystem 2 into the disk 21' in the new storage subsystem 2'. The table 100 includes a flag 102 for each logical block address 101. In the present embodiment, the case where the flag 102 is "1" indicates that the data at the address has been copied already from the disk 21 into the disk 21', and the case where the flag 102 is "0" indicates that the data at the address has not been copied yet. In the data migration processing, or in the data reading/writing processing based on the requests from the host computer during the data migration processing, the unit 24 makes reference to this table.

Referring to FIG. 3, the explanation will be given below concerning the flow of the migration processing performed by the data migration unit 24.

First, a counter B is prepared, and the initial value is set to be "0" (step 2001). Next, reference is made to the table 100, thereby checking whether or not the flag 102 for LBA B is "1" (step 2002). If the flag is "1", it means that the data migration is over, and accordingly the counter B is incremented by "1" (step 2005). Also, at the step 2002, if the flag 102 is "0", the data is copied from the disk 21 into the disk 21' (step 2003). Then, the corresponding flag 102 in the table 100 is updated to "1" (step 2004), and the processing goes to the step 2005. At a step 2006, it is checked whether or not the processing has been executed to the final LBA of the disk 21. Namely, it is checked whether or not B has exceeded the final LBA of the disk 21. If B has exceeded the LBA, the processing is completed. If not, the processing goes back to the step 2002, then being repeated.

Next, referring to FIG. 4, the explanation will be given below regarding a processing in the case where, while the data migration unit 24 is executing the data migration processing in FIG. 3, there has arrived the writing request from the host computer, i.e., the back end server 3 (or the host 1) in the present embodiment. In the case where the block unit of the written-in data from the host coincides with the unit of LBA, this processing is simple: At a step 2101, the data is written into the disk 21' in the new storage subsystem 2', and at a step 2102, the flag 102

for the corresponding LBA in the table 100 is updated to "1". Namely, with respect to the LBA for which the writing processing has been performed during the data migration processing, it becomes unnecessary to execute the data migration from the disk 21 in the old storage subsystem 2.

Referring to FIG. 5, the explanation will be given below concerning a processing in the case where, while the data migration unit 24 is executing the data migration processing in FIG. 3, there has arrived the reading request from the host computer, i.e., the back end server 3 in the present embodiment. At a step 2201, reference is made to the flag 102 for the LBA within the table 100 for which there has arrived the reading request. At a step 2202, it is checked whether or not the flag 102 for the corresponding LBA is "1", thereby branching the processing. If the flag is "1", the data migration from the disk 21 in the old storage subsystem 2 into the disk 21' in the new storage subsystem 2' has been finished with respect to the LBA, and accordingly the data is read from the disk 21' (step 2203). If the flag 102 is "0", the data migration has been not finished yet with respect to the LBA, and accordingly the data is copied from the disk 21 into the disk 21' once (step 2205). Subsequently, the flag 102 in the table 100 is updated to "1" (step 2206). After that, the processing goes to the step 2203 or after. At a step 2204, the read-out data is



passed over to the back end server 3, thereby completing the processing.

In this way, it turns out that, even during the data migration, the back end server 3 always

5 executes the data reading/writing with respect to the disk 21' in the new storage subsystem 2'.

Next, concerning the data migration processing from the old storage subsystem 2 to the new storage subsystem 2' in the system of the present embodiment, the flow of the entire system will be explained. When executing the data migration, the user instructs the back end server 3 to start the migration. The data-migration-processing starting instruction from the back end server 3 to the new storage subsystem 2' is transmitted to the new storage subsystem 2' via the interface 32.

FIG. 6 explains the flow of a virtual disk's setting modification processing executed by the back end server 3. If the back end server 3 receives the migration instruction, at first, the server 3 stops the operation of the virtual disk by the virtual device driver 31 (step 1001). This interrupts the access from the virtual device driver 31 to the old storage subsystem 2. As a result, even if the virtual device driver 31 receives an access command to the virtual disk from the host 1, the driver 31 returns no response until the access halt is cleared. Next, a storage apparatus management program 34 instructs the new

storage subsystem 2' to start the data migration processing illustrated in FIG. 3 (step 1002). Moreover, at a step 1003, the setting of the virtual disk that the virtual device driver 31 has caused the host 1 to access the disk 21 so far is modified so that an access to the disk 21' will be made. Finally, at a step 1004, the access halted at the step 1001 is restarted. When the access to the virtual disk from the host 1 is restarted, if accesses have been made to the virtual disk from the host 1 between the step 1001 and the step 1002, the accesses are all carried out with respect to the disk 21'.

Also, in the present embodiment, there has been provided the connection configuration where the old and the new storage subsystems 2 and 2' and the back end server 3 are directly connected to the switch. However, in the case where, unlike the example in FIG. 1, the additionally-installed new storage subsystem 2' does not include the data migration unit 24, the data migration unit 24 is included on the side of the back end server 3 as is illustrated in FIG. 7. Then, the back end server is caused to execute the data migration processing, which makes it possible to implement the same processing.

Moreover, in the present embodiment, there has been provided the back end server 3, thereby presenting the processing for making the host see the virtual disk. As illustrated in FIG. 8, however, it is

also possible to implement a configuration where the virtual device driver 31, the storage apparatus management program 34, and the data migration unit 24 are included in the fibre channel switch 4. Also, as illustrated in FIG. 9, it is also possible to implement even a configuration where the storage subsystems 2 and 2' are connected to each other via an interface 37 of the back end server 3. In that case, the data migration unit 24 may also be included on the side of the back end server 3.

Although, in the present embodiment, there has been presented the example where the data migration among the disks can be executed transparently to the host computer, there exist a variety of examples to which the embodiment is applicable. If the virtual device driver causes the virtual disk to correspond to the real storage apparatuses, it does not matter at all to the host computer which storage apparatus actually stores the data. This makes the embodiment applicable to the following systems, for example: A storage apparatus management system where a virtual disk with a necessary minimum capacity is defined usually, and a virtual disk with a necessary capacity can be prepared dynamically at a point-in-time when it becomes required, and a system that, depending on the data access frequency, migrates the data transparently to the host and dynamically from a low-rate disk to a high-rate disk.

WHAT IS CLAIMED IS:

1. A computer system, comprising:  
host computers,  
a plurality of storage apparatuses, a unique ID that is unchangeable from outside being assigned to each of said storage apparatuses,  
a switch for interconnecting said host computers with said plurality of storage apparatuses, and  
a back end server connected to said host computers through said switch for managing said plurality of storage apparatuses so as to provide a virtual storage apparatus to said host computers, wherein said back end server dynamically modifies a storage apparatus from an arbitrary storage apparatus of said plurality of storage apparatuses to another storage apparatus thereof, said storage apparatus being caused to disguise said virtual storage apparatus.
2. The computer system as claimed in claim 1, wherein, while dynamically modifying said storage apparatus caused to disguise said virtual storage apparatus, said back end server makes no response to an access request made from said host computers to said virtual storage apparatus.
3. The computer system as claimed in claim 1, further comprising:  
a data migration unit for migrating data in said arbitrary storage apparatus into said another

storage apparatus on a fixed-sized data block (LBA) basis, and

a table holding a flag for indicating a data migration state on said fixed-sized data block basis, wherein, in response to a writing request for said fixed-sized data block from said back end server, said data migration unit writes said data block into a corresponding storage position in said another storage apparatus, and after that, modifies a migration state of a flag to "data migration completed", said flag corresponding to said written-in data block in said table.

4. The computer system as claimed in claim 1, further comprising:

a data migration unit for migrating data in said arbitrary storage apparatus into said another storage apparatus on a fixed-sized data block (LBA) basis, and

a table holding a flag for indicating a data migration state on said fixed-sized data block basis, wherein, in response to a reading request for said fixed-sized data block from said back end server, said data migration unit

obtains a migration state of a flag by making reference to said table, said flag corresponding to said data block for which said reading request has been made, and, if said migration state is "data migration uncompleted",

migrates said corresponding data block from said arbitrary storage apparatus into said another storage apparatus, and

modifies said migration state of said flag to "data migration completed", said flag corresponding in said table to said data block for which said reading request has been made, and after that,

passes, to said back end server, said data block for which said reading request has been made.

5. The computer system as claimed in claim 1, wherein said switch is a fibre channel.

6. The computer system as claimed in claim 1, wherein said another storage apparatus has a data migration unit.

7. The computer system as claimed in claim 1, wherein said back end server has a data migration unit.

8. The computer system as claimed in claim 1, wherein said back end server and a data migration unit are built in said switch.

9. The computer system as claimed in claim 8, wherein said switch is a fibre channel.

10. A storage apparatus switching method in a computer system which comprises host computers, a plurality of storage apparatuses, a unique ID that is unchangeable from outside being assigned to each of said storage apparatuses, and a switch for inter-connecting said host computers with said plurality of storage apparatuses, said storage apparatus switching

method comprising the steps of:

causing at least one of said plurality of storage apparatuses to disguise a virtual storage apparatus so as to provide said virtual storage apparatus to said host computers, and modifying at least one of said storage apparatuses dynamically from an arbitrary storage apparatus to another arbitrary storage apparatus, at least one of said storage apparatuses being caused to disguise said virtual storage apparatus.

11. The storage apparatus switching method as claimed in claim 10, wherein, while at least one of said storage apparatuses caused to disguise said virtual storage apparatus is being dynamically modified, no response is made to an access request made from said host computers to said virtual storage apparatus.

12. The storage apparatus switching method as claimed in claim 10, wherein said switch is a fibre channel.

13. A storage apparatus's data migrating method in a computer system which comprises host computers, a plurality of storage apparatuses, a unique ID that is unchangeable from outside being assigned to each of said storage apparatuses, and a switch for inter-connecting said host computers with said plurality of storage apparatuses, said storage apparatus's data migrating method comprising the steps of:

causing at least one of said plurality of storage apparatuses to disguise a virtual storage apparatus so as to provide said virtual storage apparatus to said host computers,

creating a table holding a flag for indicating a data migration state on a fixed-sized data block (LBA) basis,

prohibiting a response to an access request made from said host computers to said virtual storage apparatus,

modifying dynamically an arbitrary storage apparatus to another arbitrary storage apparatus as at least one of said storage apparatuses caused to disguise said virtual storage apparatus,

restarting said response to said access request made from said host computers to said virtual storage apparatus,

migrating said data block into said another arbitrary storage apparatus on said fixed-sized data block basis, said data block being stored into said arbitrary storage apparatus, and

executing, toward said another arbitrary storage apparatus, an access to said data block to be performed in response to said access request made from said host computers.

14. The data migrating method as claimed in claim

13, further comprising the steps of:

writing said fixed-sized data block into a

corresponding storage position in said another arbitrary storage apparatus in response to a writing request for said fixed-sized data block from said host computers, and after that,

modifying a migration state of a flag to "data migration completed", said flag corresponding to said written-in data block in said table.

15. The data migrating method as claimed in claim 13, further comprising the steps of:

obtaining, in response to a reading request for said fixed-sized data block from said host computers, a migration state of a flag by making reference to said table, said flag corresponding to said fixed-sized data block for which said reading request has been made, and, if said migration state is "data migration uncompleted",

migrating, from said arbitrary storage apparatus into said another arbitrary storage apparatus, said data block for which said reading request has been made,

modifying said migration state of said flag to "data migration completed", said flag corresponding in said table to said data block for which said reading request has been made, and after that,

sending said read-out data block to said host computers.

16. The data migrating method as claimed in claim 13, wherein said switch is a fibre channel.

17. A computer system, comprising:  
a plurality of computers,  
a plurality of storage apparatuses, and  
a switch for interconnecting said plurality of computers with said plurality of storage apparatuses, said computer system having means for providing a virtual storage apparatus to said plurality of computers, said virtual storage apparatus being a storage apparatus that is in a correspondence with at least one of said plurality of storage apparatuses, said means for providing said virtual storage apparatus modifying said correspondence dynamically.

18. The computer system as claimed in claim 17, wherein, when said means for providing said virtual storage apparatus has modified said correspondence dynamically, said means prevents said plurality of computers from seeing that said correspondence has been changed.

## ABSTRACT OF THE DISCLOSURE

In the computer system where the plurality of host computers and the plurality of storage apparatuses are interconnected, in order to carry out the data migration between the storage apparatuses transparently to the host computers, the back end server provides the virtual disk to the host computers. From the host computers, at first, the virtual disk looks the same as the old storage subsystem. When migrating the data from the old storage subsystem to the new storage subsystem, at first, the back end server instructs the data migration unit to execute the data migration processing. Subsequently, the back end server switches the setting of the virtual disk, thereby causing the virtual disk to correspond to the new storage subsystem.

1/9

FIG. 1

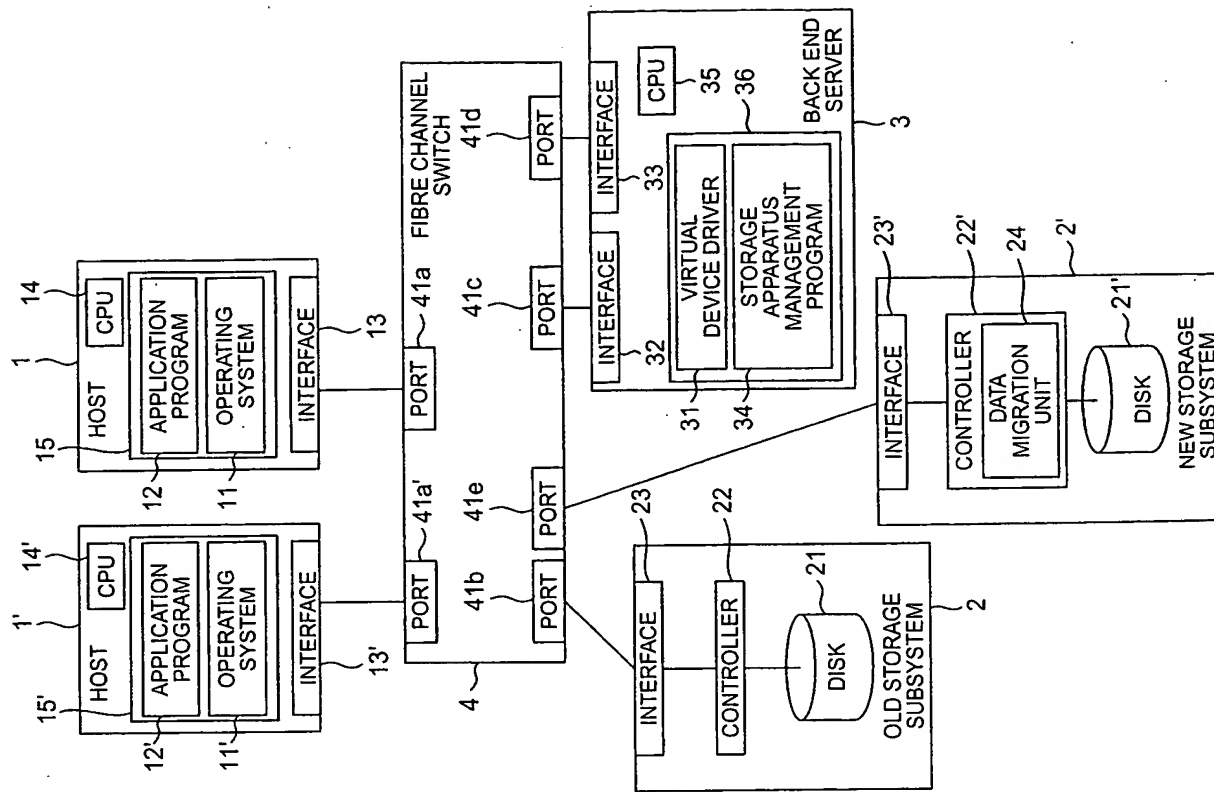


FIG. 2

100	101	102
ADDRESS	LBA	FLAG
LBA0	1	1
LBA1	1	1
LBA2	0	0
LBA3	0	0
LBA4	1	1
...	...	...
LBA <sub>n</sub>	0	0

FIG. 3

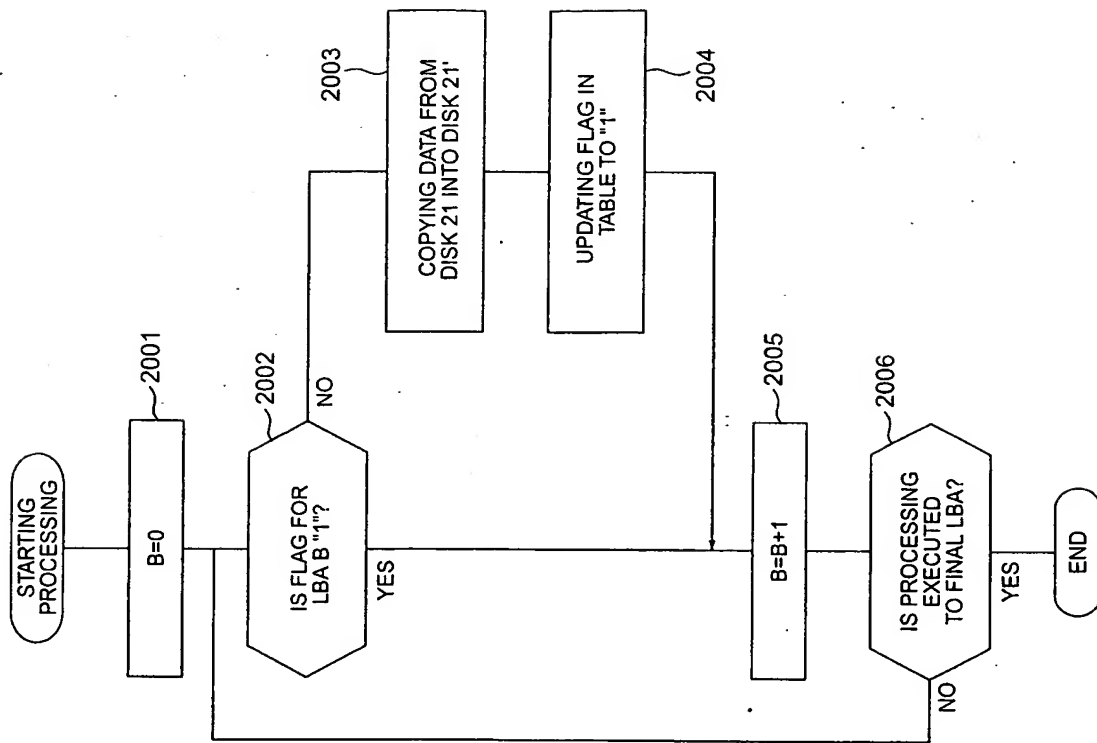


FIG. 4

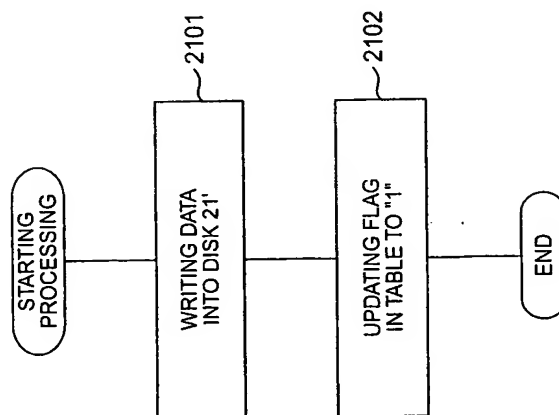


FIG. 5

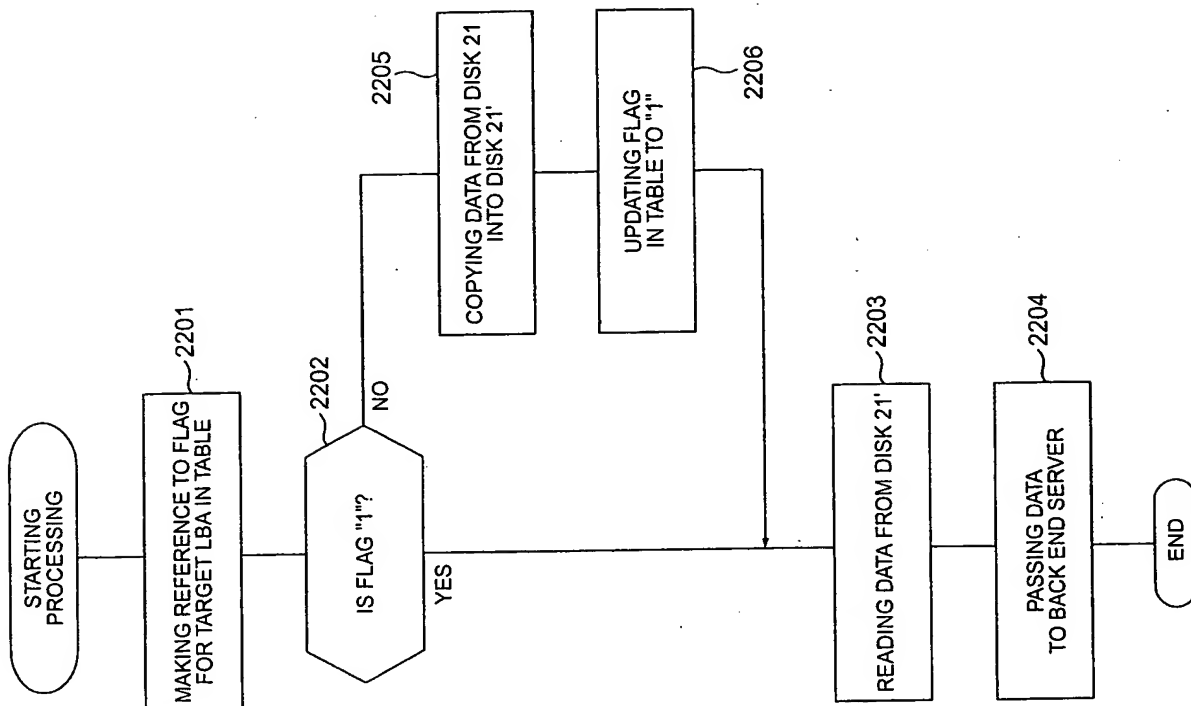




FIG. 6

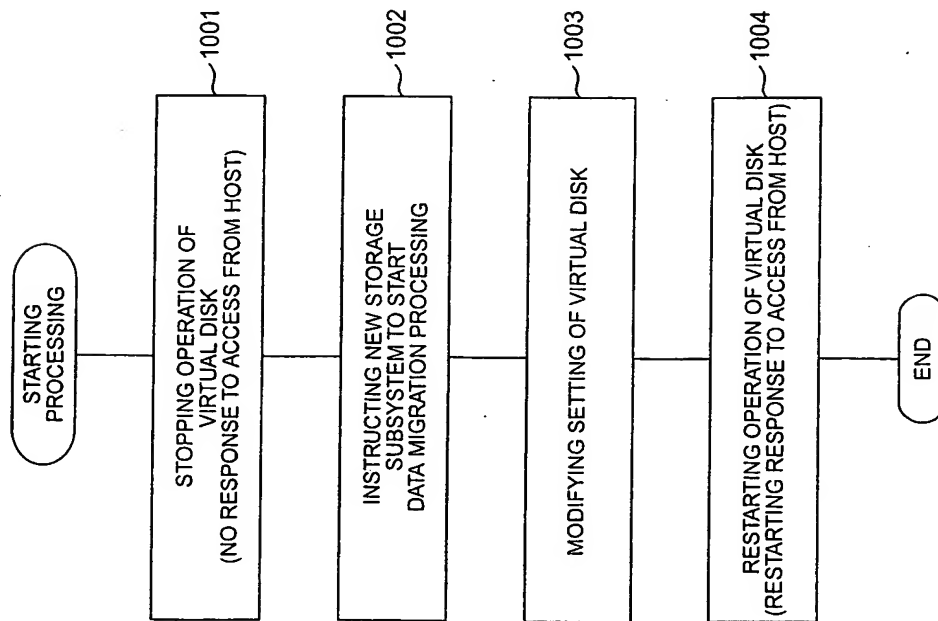


FIG. 7

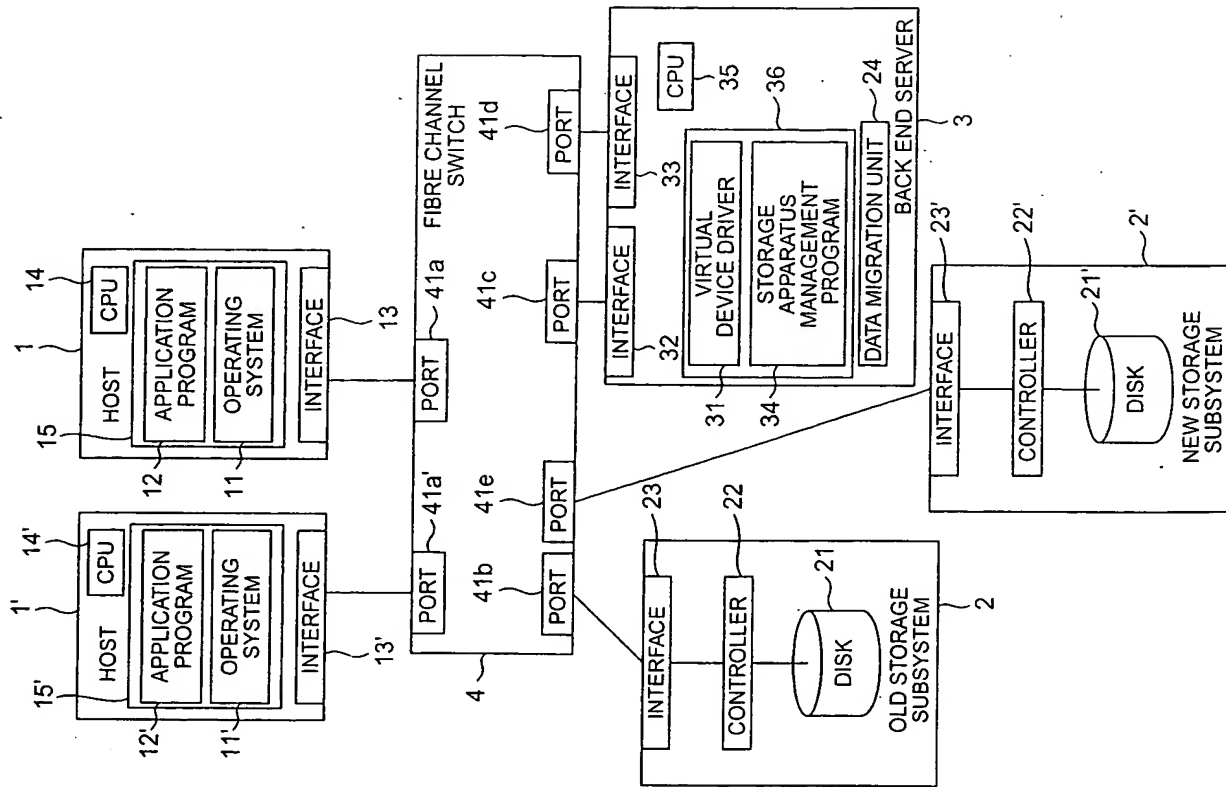


FIG. 8

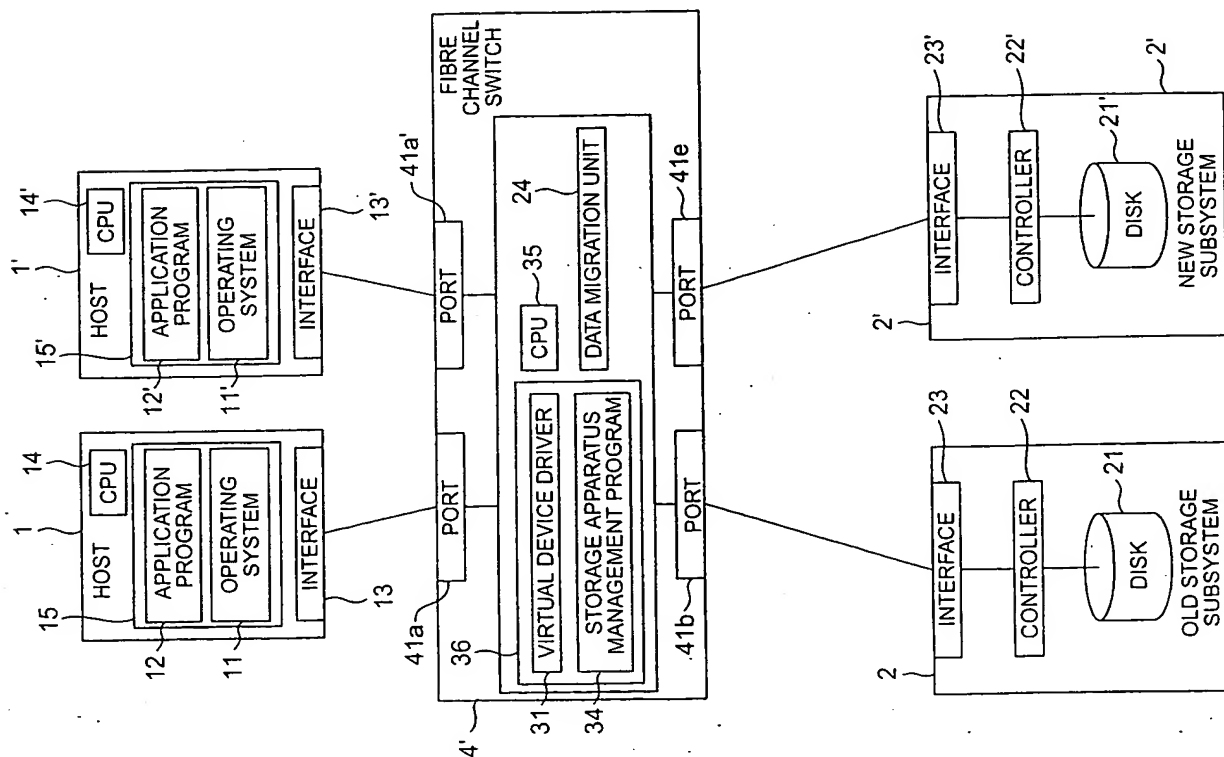
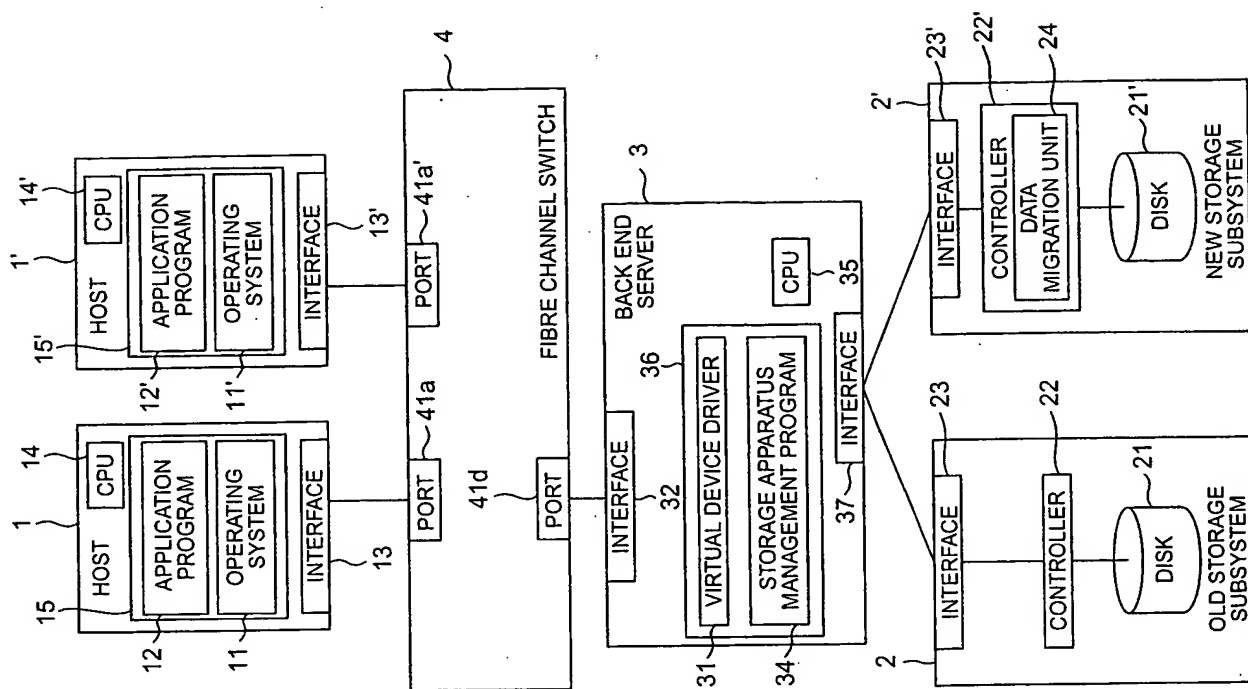
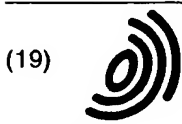


FIG. 9





Europäisches Patentamt  
European Patent Office  
Office européen des brevets



(11) EP 1 130 514 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:  
05.09.2001 Bulletin 2001/36

(51) Int Cl.7: G06F 11/14

(21) Application number: 00119434.9

(22) Date of filing: 14.09.2000

(84) Designated Contracting States:  
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU  
MC NL PT SE  
Designated Extension States:  
AL LT LV MK RO SI

(72) Inventors:  
• Watanabe, Naoki, Int. Prop. Gp., Hitachi, Ltd.  
Chiyoda-ku, Tokyo 100-8220 (JP)  
• Takamoto, Yoshifumi, Int. Prop. Gp., Hitachi, Ltd.  
Chiyoda-ku, Tokyo 100-8220 (JP)

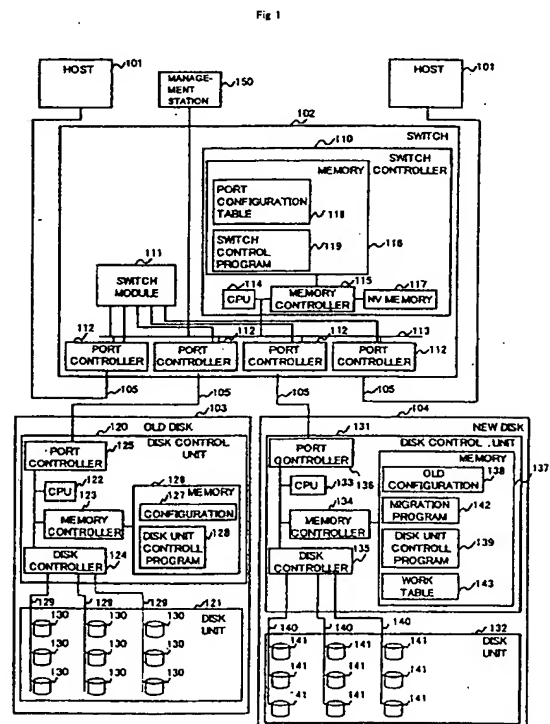
(30) Priority: 03.03.2000 JP 2000063289

(74) Representative: Beetz & Partner Patentanwälte  
Steinsdorfstrasse 10  
80538 München (DE)

(71) Applicant: Hitachi, Ltd.  
Chiyoda-ku, Tokyo 101-8010 (JP)

(54) Data migration method using storage area network

(57) The invention relates to a method of migrating data, wherein a new disk system is connected to a switch that has been already connected to a host (101) and old disk system (103). At that time, the new disk system is assigned in advance a port ID of the same value as the port ID that has been assigned to the F\_Port of the switch, so that the new disk system will not be recognized from host (101). The new disk system reads the configuration information of the old disk system. Then, in the switch, physical port ID (602) of old disk system (103) and physical port ID (602) of new disk system (104) are exchanged. The new disk system constructs volumes corresponding to the logical volume numbers and sizes of those volumes in the old disk system, and stores data of the old disk system in the corresponding volumes.



EP 1 130 514 A2

## Description

### BACKGROUND OF THE INVENTION

#### FIELD OF THE INVENTION

[0001] The present invention relates to a method of additionally attaching a new disk system (migration destination disk system) to a storage area network (hereafter referred to as SAN) and migrating data from an old disk system (migration source disk system), which is already connected to the SAN, to that new disk system and a device thereof.

#### DESCRIPTION OF RELATED ART

[0002] Conventional computer systems are configured with a plurality of servers connected by a network such as a LAN and disk units are directly connected to each computer. The data inside each disk unit was managed by the directly connected server. In other words, data was managed in a distributed condition.

[0003] However, the SAN which adopts a configuration in which a plurality of servers, a plurality of disk units, backup units, etc. are connected by a switch or hub has recently become a hot topic. Fiber channels are used to physically connect these units to a switch or hub. The advantages of configuring a system using a SAN are superior scalability and the realization of reduced management cost since unified management can be performed by integrating data distributed over a plurality of disk units. Consequently, a SAN is well suited to a large-scale system. Because many resources are interconnected with a SAN, the management of these resources is an important topic. The management of a logical volume (such as backup), as the management of a disk unit, is one example of that. At present, these management methods are being studied by the SNIA (Storage Networking Industry Association) and NSIC (National Storage Industry Consortium).

[0004] On the other hand, US patent no. 5,680,640 discloses a system and method of migrating data from a first data storage system to a second data storage system. A summary of that is as follows. A first data storage system already connected to a host, network or other data processing system is detached from the aforementioned host and connected to a second data storage system. The aforementioned second data storage system is connected to the aforementioned host or aforementioned data processing system. The aforementioned second data storage system contains a data map or data table that indicates which data element is stored in the aforementioned second storage system and which data in the aforementioned first data storage system has been copied to the aforementioned second storage system. When the aforementioned host, aforementioned network or aforementioned data processing system issues a request for data to the aforementioned second

storage system, the aforementioned second data storage system judges whether that data has been stored in the aforementioned second data storage system or in the aforementioned first storage system. If the data has been stored in the aforementioned second data storage system, that data can be used by the host, etc. If the data is not stored in the aforementioned second data storage system, the aforementioned second data storage system issues a request to the aforementioned first data storage system, and that data can be used by the aforementioned host, etc. Then, the aforementioned second data storage system writes that data to its own system and updates the aforementioned data map or aforementioned data table. When the aforementioned second data storage system is not busy, it scans the aforementioned data map or aforementioned data table, determines which data of the aforementioned first data storage system has not yet been copied into its own system, and executes the copying of that data and updating of the aforementioned data map or aforementioned data table.

#### SUMMARY OF THE INVENTION

[0005] According to the data migration system and method disclosed in US patent no. 5,680,640, independent from the aforementioned host, it is possible to execute data migration from the aforementioned first data storage system to the aforementioned second data storage system.

[0006] However, in the aforementioned data migration system and method, when the aforementioned second data storage system is utilized, the procedure of detaching the aforementioned first data storage system from the aforementioned host, connecting it to the aforementioned second data storage system, and connecting the aforementioned second data storage system to the aforementioned host is necessary. Consequently, at least during the interval beginning when the aforementioned first data storage system is detached from the aforementioned host until the aforementioned second data storage system is connected to the aforementioned host and the aforementioned first storage system, the aforementioned host cannot issue I/O (Input/Output) requests. In addition, during that interval it is necessary to temporarily suspend applications or other programs on the aforementioned host that use the aforementioned first data storage system. To further suppress costs associated with data migration, it is necessary to further reduce the time when I/O requests cannot be issued and the time when applications are suspended.

[0007] In addition, in the aforementioned data migration system and method, to connect the aforementioned first data storage system to the aforementioned second data storage system, a channel must be newly provided to each data storage system.

[0008] Further, in US patent no. 5,680,640, a data migration system and method in a SAN environment are

not disclosed.

**[0009]** The object of the present invention is to provide data migration systems and methods that are well suited to a SAN environment.

**[0010]** An overview of the typical data migration of the present invention is as follows. A host computer and first disk unit are each connected by means of a communication channel to a switch port, and it is assumed that the aforementioned first disk unit receives read/write requests from the aforementioned host computer via the aforementioned switch. Each port of the aforementioned switch is assigned a physical port ID and a logical port ID. The aforementioned switch is provided with a table that maintains the correspondence relation between the physical port ID and the logical port ID. The second disk unit is connected to a port of the aforementioned switch that is different from the port connected to the aforementioned host computer and the port connected to the aforementioned first disk unit. Via the aforementioned switch, the aforementioned second disk unit obtains configuration information (the number of logical volumes and size of each logical volume, for example) of the aforementioned first disk unit. Then, the correspondence relation between the physical port ID and the logical port ID assigned to the port of the switch connected to the aforementioned first disk unit is exchanged with the correspondence relation between the physical port ID and the logical port ID assigned to the port of the switch connected to the aforementioned second disk unit. Concretely, the logical port ID assigned to the port of the switch connected to the aforementioned first disk unit and the logical port ID assigned to the port of the switch connected to the aforementioned second disk unit are exchanged. Consequently, even if the host computer attempts to access the aforementioned first disk unit, in actuality, the aforementioned second disk unit will be accessed. After this port ID switching process is performed, the aforementioned second disk unit is configured with the logical volume that corresponds to the configuration information of the aforementioned first disk, and data in the aforementioned first disk unit is migrated to the aforementioned second disk unit. If there is a read or write request from the aforementioned host computer for data already migrated to the aforementioned second disk unit, that processing is performed for that data by the aforementioned second disk unit. If there is a read or write request from the aforementioned host computer for data that has not yet migrated to the aforementioned second disk unit, that data is read from the aforementioned first disk unit into the aforementioned second disk unit and the processing is performed for that data by the aforementioned second disk unit.

**[0011]** Other data migration methods provided by this application are clarified with the preferred embodiments of the invention.

## BRIEF DESCRIPTION OF THE DRAWINGS

**[0012]** FIG. 1 is a diagram that describes the configuration of the computer system in the first preferred embodiment of the present invention.

**[0013]** FIG. 2 is a diagram that describes the configuration of the host.

**[0014]** FIG. 3 is a diagram that describes the fabric configured with one switch.

**[0015]** FIG. 4 is a diagram that describes the fabric configured with a plurality of switches.

**[0016]** FIG. 5 is a diagram that describes the frame of the fiber channel.

**[0017]** FIG. 6 indicate an example configuration of the port configuration table of the present invention: FIG. 6 (A) indicates an example configuration of the port configuration before port switching and FIG. 6 (B) indicates an example configuration of the port configuration after port switching.

**[0018]** FIG. 7 indicates an example configuration of the old configuration information of the present invention.

**[0019]** FIG. 8 indicates an example structure of the migration worktable of the present invention.

**[0020]** FIG. 9 is a flowchart of the data migration procedure of the present invention.

**[0021]** FIG. 10 is a flowchart of the port switching procedure of the present invention.

**[0022]** FIG. 11 is a flowchart of the data migration procedure of the present invention.

**[0023]** FIG. 12 is a diagram that describes the configuration of the computer system in the second preferred embodiment of the present invention.

**[0024]** FIG. 13 is a diagram that describes the configuration of the computer system in the third preferred embodiment of the present invention.

**[0025]** FIG. 14 indicates an example configuration of the port configuration table of the third preferred embodiment of the present invention.

**[0026]** FIG. 15 indicates a flowchart of the data migration procedure of the third preferred embodiment of the present invention.

**[0027]** FIG. 16 is a diagram that describes the configuration of the computer system in the fourth preferred embodiment of the present invention.

**[0028]** FIG. 17 indicates an example configuration of the migrator of the present invention.

**[0029]** FIG. 18 is a diagram that describes the configuration of the computer system in the fifth preferred embodiment of the present invention.

**[0030]** FIG. 19 is a diagram that describes the configuration of the host in the fifth preferred embodiment of the present invention.

**[0031]** FIG. 20 is a diagram that describes the correspondence between the physical port ID and logical port ID assigned to each port prior to the port switching procedure of the present invention.

**[0032]** FIG. 21 is a diagram that describes the corre-

spondence between the physical port ID and logical port ID assigned to each port after the port switching procedure of the present invention.

[0033] FIG. 22 is an example configuration of the case where the management station of the present invention is connected to a LAN.

## DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0034] Below, preferred embodiments 1 through 5 propose data migration methods and devices, well suited to a SAN environment.

### Preferred Embodiment 1

[0035] In the present preferred embodiment, a new disk system is connected to a switch that is already connected to a host and an old disk system, and using that switch, data on the old disk system is migrated to the new disk system. Therefore, because with the present preferred embodiment, there is no need to perform the task of detaching the connection between the host and old disk system and connecting a new disk system to the host and old disk system, and it is only required to connect a new disk system to the switch, compared to the data migration system and method disclosed in the aforementioned US patent no. 5,680,640, the work of adding a new disk system is reduced. Thus, the costs associated with data migration can also be suppressed. In addition, since it is unnecessary to provide a dedicated channel for data migration with the present preferred embodiment, hardware costs can also be reduced.

[0036] Below, the present preferred embodiment will be described using FIG. 1 through FIG. 10.

### (1) System Configuration

[0037] FIG. 1 is a diagram that describes the first preferred embodiment of a computer system to which the data migration system and method of the present invention is applied. Two hosts 101, old disk system 103 and new disk system 104 are connected with switch 102, and these form a SAN. Hosts 101, old disk system 103 and new disk system 104 are connected to switch 102 with fiber channel 105. Since the present invention performs data migration from old disk system 103 to new disk system 104, there is no limitation on the number of hosts.

[0038] FIG. 5 indicates the frame format that is used with fiber channel 105. Between SOF (Start Of Frame) 501 that indicates the beginning of the frame and EOF (End Of Frame) 505 that indicates the end, frame header 502, data field 503 and CRC (Cyclic Redundancy Check) 504 are inserted into the frame. Frame header 502 includes frame control information and contains receive side address D\_ID (Destination ID) 507, sending source address S\_ID (Source ID) 508, R\_CTL (Routing

Control) 506 that controls the routing, TYPE 510 that indicates the data structure, F\_CTL (Frame Control) 511 that controls the frame sequence and exchange, SEQ\_ID (Sequence\_ID) 512 that distinguishes between the sending source and receive side sequences, SEQ\_CNT (Sequence count) 514 that indicates the count value of a number of frames of each sequence, and data field control information DF\_CNTL (Data Field Control) 513. Details of the frame of fiber channel 105 are shown in ANSI X3.230 FC-PH (Fiber Channel Physical and Signaling Interface).

[0039] FIG. 3 indicates a logical network connection configuration of the computer system shown in FIG. 1. In this computer system, fabric 301 is configured with one switch 102. Port 303 of switch 102, used in the connection of host 101 and the disk units, is called the F (Fabric) port. Also, ports 302 of host 101 and the disk units are called N (Node) ports. As indicated in FIG. 4, fabric 301 can also be configured with a plurality of switches 102. The switch ports that connect to other switches are called the E (Expansion) ports 401.

[0040] In the case where the fabric is configured with a single switch 102 as in FIG. 3, data transfers from host 101 to new disk system 104 are performed as follows. Host 101 stores its own port ID in S\_ID 508, stores the port ID of a frame's sending destination in D\_ID 507, and then sends the frame. When switch 102 receives the frame, it examines the port ID which has been stored in D\_ID 507 and sends the frame from the F\_Port which matches that port ID to the N\_Port of new disk system 104. S\_ID 508 and D\_ID are logical port IDs.

[0041] Further, as shown in FIG. 4, in the case where fabric 301 is configured from a plurality of switches 102, data transfers from host 101 to new disk system 104 are performed as follows. When switch 102 that is directly connected to host 101 receives a frame from host 101, it examines the port ID which has been stored in D\_ID 507. However, because that switch does not have the F\_Port which matches that port ID, it sends the frame from an E\_Port to another switch. Next, the switch that has the F\_Port which matches that port ID and is directly connected to new disk system 104 sends the frame from that F\_Port to the N\_Port of new disk system 104. The data transfers from host 101 to new disk system 104 have been described above as an example, but data transfers are performed in the same manner among other devices.

[0042] Again, referencing FIG. 1, the configurations of switch 102, old disk system 103, new disk system 104 and host 101 will be described in this order.

### (A) Switch 102 Configuration

[0043] Switch 102 includes switch controller 110, switch module 111 and port controller 112.

[0044] Switch controller 110 performs control within switch 102 and includes CPU 114, memory controller 115, memory 116 and nonvolatile memory 117.

[0045] CPU 114, using internal bus 113, performs the exchange of control information and data with each controller.

[0046] The read/write access to nonvolatile memory 117 is controlled by memory controller 115, and it stores such data as switch control program 119 which is necessary for the control of switch 102 and port configuration table 118 which is necessary during execution of that program.

[0047] FIG. 6 indicates an example configuration of port configuration table 118. FIG. 6 (A) shows a port configuration table before port switching, and FIG. 6 (B) shows a port configuration table after port switching. The port switching procedure will be described later.

[0048] Port configuration table 118 indicates the port configuration, and includes logical port ID 601 that indicates the logical port ID and physical port ID 602 that indicates the physical port ID. In addition, it may also include the rule of the transfer service class of the fiber channel and the port type such as loop. In the present preferred embodiment, as indicated in FIG. 20, it is assumed that a host 101 is connected to logical port ID\_0, old disk system 103 is connected to logical port ID\_1, and a host 101 is connected to logical port ID\_3. Also, it is assumed that new disk system 104 is connected to logical port ID\_2.

[0049] The read/write access to memory 116 is controlled by memory controller 115. When switch control program 119 is executed by CPU 114, that program is read from nonvolatile memory 117 and stored in memory 116. Further, at that time and as necessary, CPU 114 also reads port configuration table 118 from nonvolatile memory 117 and stores it in memory 116.

[0050] Port controller 112 performs control of the data coding/compounding. Concretely, port controller 112 extracts serial data from a received fiber channel frame and converts it into parallel data, and extracts information necessary for switching such as the send destination ID from the frame and sends it to switch module 111. Also, port controller 112 performs the reverse procedure.

[0051] Switch module 111 has a plurality of ports that are individually connected to a plurality of port controllers 112. Switch module 111 is connected to CPU 114 via internal bus 113 and is controlled by CPU 114. When switch module 111 receives data from port controller 112, in accordance with the send destination ID of that data, it switches a port to the output port and sends data. Switch module 111 may be made with a crossbar switch.

[0052] Management station 150, connected to switch 102, sets the various parameters of switch 102, and performs control of the migration process which is the most characteristic function of the present preferred embodiment and information management. Further, management station 150 has the command to active the migration program (to be described later) on new disk system 104 and the command to acquire migration information (such as migrating, done, or error).

[0053] In FIG. 1, management station 150 is directly connected to switch 102. However, as shown in FIG. 22, the aforementioned settings may be implemented with the Web or other means, by connecting host 101, switch 102, old disk system 103, new disk system 104 and management station 150 to a LAN (Local Area Network).

(B) Old disk system 103 and New disk system 104 Configurations

[0054] In present preferred embodiment, in order to simplify the description, old disk system 103 and new disk system 104 are assumed to have the same configuration, except for programs and information in memory.

[0055] Old disk system 103 includes disk control unit 120 and disk unit 121.

[0056] Disk unit 121 includes a plurality of disk drives 130. These disk drives 130 are connected to disk controller 124 with interface (I/F) 129 (fiber channel, SCSI, etc.). In the implementation of the present invention, the number of disk drives 130 is not restricted.

[0057] Disk controller 120 includes CPU 122, memory controller 123, port controller 125, disk controller 124 and memory 126. CPU 122, memory controller 123, port controller 125 and disk controller 124 are connected with an internal bus.

[0058] CPU 122, using the internal bus, performs the exchange of control information and data with these controllers. CPU 122 issues read and write commands to disk drive 130 that is necessary in the processing of commands sent from host 101 via switch 102. CPU 122 constructs the well-known RAID 0-5 configuration with a plurality of disk drives 130, and provides logical volumes for host 101.

[0059] Port controller 125 is connected to port controller 112 of switch 102 by fiber channel 105 and has the same functions as port controller 112.

[0060] Memory 126 is connected to memory controller 123 and stores disk unit control program 128 that controls the disk unit, data that are necessary during execution of said program, and configuration information 127. Configuration information 127 will be described later. Disk unit control program 128 is executed by CPU 122 and is a program that controls port controller 125 and disk controller 124, and processes read/write commands received from host 101.

[0061] As described above, except for the program and information in memory 137, the configuration elements of new disk system 104 are the same as the configuration elements of old disk system 103. Therefore only the program and information in memory 137 will be described. Stored in memory 137 is old configuration information 138 which has been read and transferred from configuration information 127 stored in memory 126 of old disk system 103, migration program 142 used during data migration, and migration worktable 143 that indicates the state of the data migration.

[0062] FIG. 8 indicates an example structure of migration worktable 143. Migration worktable 143 includes volume number 801, slot number 802 and status 803 that indicates the status of said slot.

#### (C) Host 101 Configuration

[0063] FIG. 2 indicates the configuration of host 101. Host 101 includes CPU 201, memory controller 202, port controller 206, disk controller 204, disk drive 205 and memory 203.

[0064] CPU 201, memory controller 202, port controller 206 and disk controller 204 are connected with internal bus 207. CPU 201, using internal bus 207, performs the exchange of control information and data with these controllers.

[0065] Port controller 206 is connected to port controller 112 of switch 102 with fiber channel 105 and performs the exchange of commands and data with port controller 112.

[0066] Disk drive 205 is connected to disk controller 204 and stores operating system 208, device driver 209 that controls hardware such as port controller 206, application program 210 and data that is necessary during execution of these programs.

[0067] Memory 203 is connected to memory controller 202. When operating system 208, disk controller 204, device driver 209, application program 210, etc. are executed by CPU 201, they are read from disk drive 205 and stored in memory 203.

#### (2) Data Migration Procedure Flow

[0068] Next, referencing FIG. 9, the data migration procedure of the present invention will be described. It is assumed that host 101 and old disk system 103 have been already connected to switch 102.

##### (A) Connect New Disk System to Switch (901)

[0069] First, an operator activates management station 150 and then inputs the port numbers of switch 102 that are connected to old disk system 103 and new disk system 104.

[0070] Next, the operator connects new disk system 104 to switch 102. At this point in time, if the port ID of new disk system 104 is a port ID other than a port ID already in use, any port ID may be assigned. When each port of switch 102 is assigned a port ID by default, the port ID assigned to the F\_Port connected to new disk system 104 is appropriate. In the present preferred embodiment, as described above, logical port ID\_2 is assigned to new disk system 104.

[0071] Because the logical port ID assigned to new disk system 104 is different from logical port ID\_1 of the disk unit that can be used by host 101, at this point in time, host 101 cannot access new disk system 104.

##### (B) Transfer Configuration of Old disk system to New disk system (902)

[0072] After connecting new disk system 104, the operator activates migration program 142 on new disk system 104 with the aforementioned command from management station 150. The activated migration program 142 first obtains configuration information 127 from old disk system 103 (902).

[0073] As indicated in FIG. 7, configuration information 127 includes old port ID 702, world wide name 703, fiber configuration information 704 and SCSI configuration information 705.

[0074] Fiber configuration information 704 includes PLOGI payload 706 and PRLI payload 707.

[0075] PLOGI payload 706 includes common service parameter 708 that are exchanged at the time of a fiber channel N\_Port login (PLOGI), port name 709, node name 710 and class service parameter 711. Specified in common service parameter 708 are the fiber channel version information and functions such as the address specification method supported by said fiber channel device, the communication method, etc. Class service parameter 711 indicates the class support information, X\_ID re-assignment, ACK capability, etc. Details of fiber channel parameters are described in the aforementioned ANSI X3.230 FC-PH. PRLI payload 707 includes service parameter 712.

[0076] SCSI configuration information 705 includes inquiry data 703, disconnect/reconnect parameter 714, sense data 715 and mode select parameter 716. Inquiry data 713 indicates the type, vendor ID, product ID, etc. of the SCSI device. Disconnect/reconnect parameter 714 indicates the connection conditions. When an error occurs, sense data 715 is exchanged to investigate the state of the disk unit. Mode select parameter 716 performs the setting and modification of various parameters related to the physical attributes of the SCSI device, data format of the storage media, error recovery method and procedure, processing method of an I/O process, etc. By means of inquiry data 713 and mode select parameter 716, the number of volumes and size of each volume (number of blocks) on old disk system 103 can be learned. A detailed description of the protocol for SCSI on fiber channel 105 is shown in ANSI X.269 Fiber Channel Protocol for SCSI.

[0077] As described above, configuration information 127 can also exist as information in addition to that which can be acquired using protocols of the existing fiber channel, SCSI, etc. If that type of configuration information exists, the operator directly reads configuration information 127 from old disk system 103 and transfers it to new disk system 104. Reading and transferring are performed from an operation panel or by web access by means of HTTP.

[0078] When the transfer of configuration information 127 to new disk system 104 is completed, or in other words, when the transfer of old configuration information



138 is completed, migration program 142 of new disk system 104 notifies switch 102 of that fact.

#### (C) Port Switching Procedure (903)

[0079] When switch control program 119 of switch 102 receives that notification, the port switching procedure is started.

[0080] Details of the port switching procedure are described using the flowchart of FIG. 10.

[0081] Initially, the case as indicated in FIG. 3 where fabric 301 is configured with a single switch 102 will be described.

[0082] First, switch control program 119 of switch 102 notifies all hosts 101 using old disk system 103 that port switching has begun. Device driver 209 of host 101 that has received this notification queues an I/O process for old disk system 103 in memory 203 of host 101 (1001).

[0083] When an I/O process for old disk system 103 is suspended, device driver 209 notifies switch 102 of the I/O suspension completion. If an I/O process for old disk system 103 is being executed, it may be terminated, but it is desirable to allow the execution to finish and then issue notification of the I/O suspension completion. After receiving such notification from all hosts 101, switch control program 119 of switch 102 changes the correspondence relation between logical port ID 601 and physical port ID 602 of old disk system 103 and the correspondence relation between logical port ID 601 and physical port ID 602 of new disk system 104 of port configuration table 118 (1002). In other words, port configuration table 118 is rewritten as in FIG. 6 (B). Using diagrams of the whole computer system, this situation can be illustrated as the change from the state indicated in FIG. 20 to the state indicated in FIG. 21.

[0084] Thereafter, by means of referencing port configuration table 118 at each frame transmission or reception, port controller 112 of switch 102 performs the port switching procedure by manipulating S\_ID 508 and D\_ID 507. When a frame is received, logical port ID 601 that corresponds to S\_ID 508 of the frame is retrieved, and S\_ID 508 of the frame is converted into physical port ID 602 that corresponds to the retrieved logical port ID 601. Similarly, when a frame is transmitted, logical port ID 601 that corresponds to D\_ID 507 of the frame is retrieved, and that D\_ID 507 is converted into physical port ID 602 that corresponds to the retrieved logical port ID 601. At this time, CRC 504 that is appended to the frame is recalculated. By means of the above procedure, all frames to old disk system 103 are sent to new disk system 104. Further, frames sent from new disk system 104 are viewed by host 101 as being sent from old disk system 103. After port switching by switch 102, host 101 I/O is restarted (1003).

[0085] Next, as indicated in FIG. 4, the case where fabric 301 is configured with a plurality of switches 102 will be described.

[0086] First, switch 102 which will become the master

(master switch) is determined. In the present preferred embodiment, the master switch is switch 102 that is directly connected to new disk system 104.

[0087] Switch control program 119 of the master switch notifies all hosts 101 that are using old disk system 103 and all switches except for the master switch in fabric 301 of the starting of port switching. The processing to be performed by device driver 209 of host 101 that has received such notification is the same as in the case where the fabric is configured from a single switch 102, except for notification to the master switch of the I/O suspension completion.

[0088] After such notification is received from all hosts 101, switch control program 119 of the master switch changes the correspondence relation between logical port ID 601 and physical port ID 602 of old disk system 103 and the correspondence relation between logical port ID 601 and physical port ID 602 of new disk system 104 of port configuration table 118, and notifies all switches 102 except the master switch of that change. Based on such notification, all switches except for the master switch change their own port configuration table 118. Thereafter, the operation of each switch is the same as the operation of switch 102 in the case where the fabric is configured with a single switch 102. After port switching has been performed by all switches 102 in fabric 301, host 101 I/O is restarted.

#### (D) Data Migration Procedure (904)

[0089] The data migration procedure is performed synchronously with the completion of the port switching procedure. This data migration procedure will be described using the flowchart of FIG. 11.

[0090] First, migration program 142 constructs a volume in new disk system 104 that corresponds to the logical volume number and size of that volume in old disk system 103, and then initializes variables to be used and migration worktable 143 that was described with FIG. 8 (1101).

[0091] Migration program 142 on new disk system 104 checks whether there is an I/O request from host 101 (1102).

[0092] If there is no I/O request from host 101, data is migrated in slot units from old disk system 103. At that time, using migration worktable 143, migration program 142 computes the address of the next data to migrate (1103).

[0093] Data migration is performed for all the logical volumes of old disk system 103 in order of increasing volume number 801. Further, within each volume, data migration is performed beginning with the first block. Worktable 143 for migration shown in FIG. 8 indicates the next migration address in the volume number 801 cell of the first row. The initial value of the next migration address is the first block address of the smallest volume number 801. Moreover, migration worktable 143 indicates the next migration slot in the slot number 802 cell

of the first row. From the second row onward, the rows are arranged in order of increasing volume number 801 and slot number 802. When migration program 142 transfers data of the next migration address obtained from migration worktable 143, it changes status 803 of that slot number to "Migrating" (1104).

[0094] Next, using port controller 125, migration program 142 issues a request to read the slot size to old disk system 103 and acquires the corresponding data (1105).

[0095] Migration program 142 writes the data obtained from old disk system 103 to a corresponding volume of new disk system 104 on disk drive 141, using disk controller 124 and via I/F 140 (1106).

[0096] After the data is written on disk drive 141, it is checked whether the migration of all slots of the corresponding volume is completed. If completed, data migration of the next volume is executed. If not completed, the slot number is incremented (1107).

[0097] If the migration of all volumes is completed, the migration procedure is finished (1108).

[0098] Next, the case of an I/O request from host 101 will be described. If there is an I/O request from host 101, migration program 142 investigates whether the request is for a read or write access (1109).

[0099] In the case of a read request, migration program 142 references migration worktable 143 and investigates whether migration of the requested data has been completed (1110).

[0100] If the migration is completed, migration program 142 reads that data from disk drive 141 in new disk system 104 and sends it back to host 101 (1113, 1112).

[0101] If the migration of the data requested from host 101 is not complete, using port controller 125, migration program 142 issues a read request to old disk system 103 via switch 102 and acquires said data. Migration program 142 sends back the data acquired from old disk system 103 to host 101 and also writes that data to the corresponding volume of new disk system 104 (1111, 1112).

[0102] After said data is written on disk drive 141, said data is registered into migration worktable 143, and with status 803 set to "Done", the processing of said request is finished (1108).

[0103] Next, the case of a write request from host 101 will be described. Migration program 142 receives data related to the write request from host 101 (1114).

[0104] Then, using port controller 125, migration program 142 issues a read request in slot size units to old disk system 103 via switch 102 so as to include the data requested from host 101, and acquires the data. After migration program 142 overwrites the data obtained from the migration source with the data received from host 101 and creates new slot data, it writes to the volume in new disk system 104 requested by host 101 (1115).

[0105] After the write is completed, migration program 142 registers said slot into migration worktable 143, and

with status 803 set to "Done", the processing related to the write request from host 101 is finished (1107).

(E) Detach Old disk system (905)

[0106] After the data migration procedure is completed, the operator detaches old disk system 103 from the switch and terminates the migration procedure (905).

[0107] The system configuration and flow of the data migration procedure have been described above for preferred embodiment 1. By means of the present preferred embodiment, I/O requests from the host are only suspended during the time while configuration information of old disk system 103 is being transferred to new disk system 104 and during the time while the port switching procedure is being performed. Therefore, by means of the present preferred embodiment, because there is no need to perform the task of connecting a new disk system to the host and old disk system, by detaching the connection between the host and old disk system, the time while I/O requests from the host are suspended is shorter than the time while I/O requests from the host are suspended in the data migration system and method disclosed in US patent no. 5,680,640. Consequently, the costs associated with data migration can be suppressed. Moreover, by means of the present preferred embodiment, since there is no need to provide a dedicated channel for data migration, hardware costs can also be suppressed.

Preferred Embodiment 2

[0108] The present preferred embodiment differs from preferred embodiment 1 in that in addition to the system configuration described with preferred embodiment 1, the old disk system and new disk system are directly connected with an interface (fiber channel or SCSI, for example). In the present preferred embodiment, since data migration is performed using that interface, compared to preferred embodiment 1, high-speed data migration can be performed. In the present preferred embodiment, it is necessary to provide an interface for data migration. However, there is no need to perform the task of connecting a new disk system to the host and old disk system, by detaching the connection between the host and old disk system, and because it is only required to connect a new disk system to the switch and connect the old disk system and the new disk system with an interface, compared to the data migration system and method disclosed in the aforementioned US patent no. 5,680,640, the work of adding the new disk system is reduced.

[0109] Below, the present embodiment will be described using FIG. 12.

[0110] FIG. 12 is a diagram that describes the second preferred embodiment of a computer system that has adopted the data migration system and method of the present invention. As described above, each configura-

tion element of the computer system is the same as each configuration element of FIG. 1, and therefore their descriptions are omitted.

[0111] Below, the flow of the data migration procedure in the present preferred embodiment will be described. The data migration procedure flow is the same as that of preferred embodiment 1, which was described with FIG. 9. However, in the present preferred embodiment, since old disk system 1203 and new disk system 1204 are connected directly by a interface and data migration is performed using that interface, it is necessary to obtain the correspondence relation between the logical address and physical address in old disk system 1203 from old configuration information 1238. Below, only this point of difference will be described.

[0112] In steps 1105, 1113 and 1115 of FIG. 11, data is read from the old disk system via switch 1202. However, in the present preferred embodiment, using information that is included in old configuration information 1238, migration program 1231 computes the physical address corresponding to the logical address of the data that should be read from old disk system 1203. Then, using I/F 1240, migration program 1231 issues directly a read request to disk drive 1230 of old disk system 1203 that corresponds to the computed physical address, and acquires that data. At this time, if that data has been distributed to a plurality of disk drives 1230 of old disk system 1203, or if stored in dispersed physical addresses, a plurality of read requests will be issued.

[0113] In the present preferred embodiment, since old disk system 1203 and new disk system 1204 are connected directly with an interface, and data migration is performed using that interface without passing through switch 1202, compared to preferred embodiment 1, high-speed data migration can be performed. Further, since the data migration of the present preferred embodiment does not pass through switch 1202, the resource consumption of the switch can be suppressed. In the present preferred embodiment, two hosts and two disk units are connected to switch 1202. However, because many hosts and disk units are connected in a SAN environment, those resources can be allocated to the data transfer between other hosts and other disk units. Therefore, by means of the present preferred embodiment, compared to preferred embodiment 1, the degradation of switch performance that accompanies data migration can be suppressed.

#### Preferred Embodiment 3

[0114] The present preferred embodiment differs from preferred embodiment 1 in that the migration program is executed by a switch. By means of the present preferred embodiment, since it is unnecessary for a disk unit to have a migration program, data migration from an old disk system is possible even with a disk unit that does not have this type of program.

[0115] Using FIG. 13, the configuration of the compu-

ter system of the present preferred embodiment will be described. However, a description of those configuration elements that are the same as configuration elements of FIG. 1 will be omitted.

[0116] In the present preferred embodiment, non-volatile memory 1317 in switch 1302 stores switch control program 1319 that is necessary for control of switch 1302, migration program 1342 that performs the migration procedure, port configuration table 1318 that is necessary during execution of these programs, old configuration information 1327 and migration worktable 1343. Disk unit control program 1339 is stored in memory 1337 in new disk system 1304, but the old configuration information, migration worktable, and migration program are not stored, though they are in preferred embodiment 1.

[0117] FIG. 14 is an example configuration of port configuration table 1318 that indicates the port configuration. Port configuration table 1318 includes logical port ID 1401 that indicates the virtual port ID, physical port ID 1402 that indicates the physical port ID, and status 1403 that indicates the port status.

[0118] Next, using the flowchart of FIG. 15, the data migration procedure of the present preferred embodiment will be described.

[0119] An operator connects new disk system 1304 to switch 1302 (1501). This step is the same as step 901 of FIG. 9.

[0120] Next, migration program 1342 of switch 1302 reads configuration information 1327 from old disk system 1303 (1502). Configuration information 1327 is the same as that shown in FIG. 7.

[0121] After configuration information 1327 of old disk system 1303 has been transferred to switch 1302, switch control program 1319 of switch 1302 starts the port switching procedure (1503).

[0122] Also in the port switching procedure of the present preferred embodiment, similar to preferred embodiment 1, after the host I/O is stopped, the correspondence relation between logical port ID 1401 and physical port ID 1402 of old disk system 1303 and the correspondence relation between logical port ID 1401 and physical port ID 1402 of new disk system 1304 of port configuration table 1318 are changed. However, the port switching procedure of the present preferred embodiment differs from that of preferred embodiment 1 in the following point. In the present preferred embodiment, switch control program 1319 of switch 1302 sets port ID status 1403 of old disk system 1303 in port configuration table 1318 to "Migrating." The frame for the port ID with "Migrating" status 1403 in port configuration table 1318 is not sent to the corresponding port, but instead is passed to migration program 1331 of switch 1302. In other words, access to old disk system 1303 is performed by migration program 1331 of switch 1302.

[0123] In the case where the fabric is configured with a plurality of switches 1302, using the same method as preferred embodiment 1, after the host I/O is stopped, switch control program 1319 of the master switch sets

port ID status 1403 of old disk system 1303 in port configuration table 1318 to "Migrating." The frame for the port ID with "Migrating" status 1403 in port configuration table 1318 is not sent to the corresponding port, but instead is passed to migration program 1331 of switch 1302. In the case where the fabric is configured with a plurality of switches 1302, it is sufficient if at least old configuration information 1327, migration worktable 1343, and migration program 1342 are provided in switch 1302, the master switch.

[0124] After the port switching procedure is completed, the data migration procedure is performed. First, migration program 1342 of switch 1302 constructs a volume according to the logical volume number and size of said volume which are configured in old disk system 1303, and then initializes variables to be used and migration worktable 1332 that is shown with FIG. 8. The procedure thereafter is the same procedure as described in preferred embodiment 1, except that it is not performed by the migration program of the new disk system, but instead by migration program 1302 of switch 1302.

[0125] After the data migration procedure is completed, switch control program 1319 of switch 1302 changes the status of new disk system 1304 in port configuration table 1318 to "Normal" (1505). The operator detaches old disk system 1303 from switch 1302 and terminates the migration procedure (1506).

#### Preferred Embodiment 4

[0126] The characteristic of the present preferred embodiment is that a migration program is executed on a migrator configured with a personal computer, workstation, etc. which are connected to a switch. By means of the present preferred embodiment, similar to preferred embodiment 3, since it is unnecessary for a new disk system to have a migration program, data migration from an old disk system is possible even with a disk unit that does not have this type of program. Further, in the present preferred embodiment, as in preferred embodiment 3, since the migration program is not executed on a switch, a load of the switch can be reduced.

[0127] Using FIG. 16, the configuration of the computer system of the present preferred embodiment will be described. However, a description of those configuration elements that are the same as configuration elements of FIG. 1 will be omitted.

[0128] A characteristic of the present preferred embodiment, as will be described later, is that migrator 1606 internally stores the migration program that performs the migration procedure, old configuration information and migration worktable. Memory 1637 in new disk system 1604 stores disk unit control program 1639, but does not store the old configuration information, migration worktable and migration program, though in preferred embodiment 1. Further, different from preferred embodiment 3, in the present preferred embodiment,

the migration program, old configuration information and migration worktable are not stored in switch 1602.

[0129] FIG. 17 indicates an example configuration of migrator 1606. Migrator 1606 includes CPU 1701, memory controller 1702, port controller 1706, disk controller 1704, disk drive 1705 and memory 1703. CPU 1701 is connected to memory controller 1702, port controller 1706 and disk controller 1704 with internal bus 1701. CPU 1701 performs the exchange of control information and data with each controller. Port controller 1706 is connected to port controller 1612 of switch 1602 with fiber channel 1605 and performs the exchange of commands and data with switch 1602. The port of migrator 1606 is called the N port. The port on the switch 1602 side is called the F port. Disk drive 1705 is connected to disk controller 1704 and receives read and write requests from CPU 1701. Disk drive 1705 stores programs necessary for migrator 1606, and stores migrator program 1731 and migrator worktable 1732 that is necessary during execution of said program. Memory 1703 is connected to memory controller 1702. During execution of migration program 1731, this program is read from drive 1705 into memory 1703.

[0130] Next, the flow of the data migration procedure of the present preferred embodiment will be described. An operator connects new disk system 1604 to switch 1602 for data migration. At this time, the port ID of new disk system 1604 takes the same value as the port ID assigned to the F\_Port of switch 1602, and new disk system 1604 is not recognized from host 1601. This step is the same as step 901 described in FIG. 9.

[0131] Next, migration program 1631 of migrator 1606 reads configuration information 1727 from old disk system 1603. Configuration information 1727 is the same as that indicated in FIG. 7.

[0132] After configuration information 1727 of old disk system 1603 is transferred to the migrator, the port switching procedure is performed. This procedure, similar to that described with preferred embodiment 1, suspends I/O from the host, and exchanges physical port ID 1604 of old disk system 1603 and physical port ID 1402 of new disk system 1604 in port configuration table 1618. The control program of switch 1602 sets the port ID status of old disk system 1603 in port configuration table 1618 to "Migrating." If the status in the port configuration table is "Migrating", the frame is not sent to a port, but instead is passed to migration program 1731 of migrator 1606. Access to old disk system 1603 is performed by migration program 1731 of migrator 1606. After these procedures, the I/O of host 1601 is restarted and migration program 1731 of migrator 1606 starts the migration procedure.

[0133] After the port switching procedure is completed, the data migration procedure is performed. First, migration program 1731 of migrator 1606 constructs a volume according to the logical volume number and size of said volume which are configured in old disk system 1603, and then initializes variables to be used and mi-

gration worktable 1732. The procedure thereafter is the same procedure as described in preferred embodiment 1, except that it is not performed by the migration program of the new disk system, but instead by migration program 1731 of migrator 1606.

[0134] After the data migration procedure is completed, switch control program 1619 of switch 1602 changes the status of new disk system 1604 in port configuration table 1618 to "Normal," as in preferred embodiment 3. The operator detaches old disk system 1603 from switch 1602 and terminates the migration procedure.

[0135] In addition, in the present preferred embodiment, a fiber channel connects the migrator and switch. However, it can be easily understood by those skilled in the art of the present invention that the effect of the present preferred embodiment can be obtained with connection by such means as a vendor-unique bus.

#### Preferred Embodiment 5

[0136] The characteristic of the present preferred embodiment is that the host implements the port switching of the switch. By means of the present preferred embodiment, there is no need to provide a port configuration table in the switch, though in preferred embodiment 1 through preferred embodiment 4. Therefore, it is possible to perform data migration even in cases where a SAN is configured using a switch that does not have this function.

[0137] Using FIG. 18, an example configuration of the computer system of the present preferred embodiment will be described. However, a description of those configuration elements that are the same as configuration elements of FIG. 1 will be omitted. As will be described later, in the present preferred embodiment, differing from preferred embodiment 1, a port configuration table is provided in host 1801 and a port configuration table is not provided in switch 1802.

[0138] FIG. 19 indicates an example configuration of host 1801. Host 1801 comprises CPU 1901, memory controller 1902, port controller 1906, disk controller 1904, drive 1905 and memory 1903. CPU 1901 is connected to memory controller 1902, port controller 1906 and disk controller 1904 with internal bus 1907. CPU 1901 performs the exchange of control information and data with each controller. Port controller 1906 is connected to port controller 1812 of switch 1802 with fiber channel 1805, and performs the exchange of commands and data with switch 1802. The port of host 1801 is called the N port, and the port on the switch 1802 side is called the F port. Drive 1905 is connected to disk controller 1904 and receives read and write commands from CPU 1901. Drive 1905 stores programs necessary for host 1801, and stores operating system 1908, device driver 1909, application program 1910 and port configuration table 1918. Memory 1903 is connected to memory controller 1902 and receives read and write requests from CPU 1901. Programs such as operating system

1908, disk controller 1904, device driver 1909 that controls the hardware, and application 1910 are read from disk drive 1905 into memory 1903 at the time of each program execution.

5 [0139] Next, the flow of the data migration procedure of the present preferred embodiment will be described. This flow is the same as the flow of the data migration procedure of preferred embodiment 1, which is shown in FIG. 9. However, in the present preferred embodiment, the port switching procedure is different from that of preferred embodiment 1. Here, only the port switching procedure will be described.

10 [0140] The port switching procedure of the present preferred embodiment performs the same operation both in the case where fabric 301 is configured with a single switch, and in the case where it is configured with a plurality of switches. At first, by operator command, device driver 1909 of the host notifies all hosts 1801 using old disk system 1803 of the start of port switching. 15 Device driver 1909 of host 1801 that has received this notification temporarily stores I/O for old disk system 1803 and then queues it in memory 203 of host 1801. On the other hand, an I/O process being executed for old disk system 1803 is executed until completed. When 20 an I/O process for old disk system 1803 is suspended, device driver 1909 notifies switch 1802 of I/O suspension completion. After receiving I/O suspension completion notification from all hosts 1801, device driver 1909 of the host that issued the notification of port switching requests all the hosts to exchange physical port ID 602 25 of old disk system 1803 and physical port ID 602 of new disk system 1804 in port configuration table 1918. Thereafter, the device driver of the host references port configuration table 1918 at each frame transmission or reception and performs the port switching procedure by manipulating S\_ID 508 and D\_ID 507. This procedure is the same as the procedure described with preferred embodiment 1. 30

35 [0141] Above, preferred embodiment 1 through preferred embodiment 5 of the present invention have been described. Appropriate combined embodiments of these preferred embodiments are possible. For preferred embodiments 3 through 5, an example would be to connect old disk system 1303 and new disk system 1304 with a disk interface for migration, similar to preferred embodiment 2, and perform data migration via this interface. 40

45 [0142] As described above, data migration methods and devices well suited to a SAN environment can be provided by means of this invention.

50 [0143] Although the present invention has been described in connection with the preferred embodiments thereof, many other variations and modification will become apparent to those skilled in the art.

# Claims

1. A method of migrating data from a first storage system that is connected to a switch (102) by means of a communication channel and receives read/write requests from a host computer (101) connected to the switch (102) by means of a communication channel, to a second storage system, comprising the steps of:

connecting the second storage system to the switch (102) by means of a communication channel; and,  
writing, via the switch (102), data stored in the first storage system to the second storage system.

2. A method of migrating data from a first storage system that is connected to a switch (102) by means of a communication channel and receives read/write requests from a host computer (101) connected to the switch (102) by means of a communication channel, to a second storage system, comprising the steps of:

connecting the second storage system to the switch (102) by means of a communication channel;  
reading the number of logical volumes constructed in the first storage system and their sizes are read into the second storage system via the switch (102);  
constructing the same number and same sizes of logical volumes as logical volumes that have been constructed in the first storage system in the second storage system; and,  
writing, via the switch (102), data stored in the first storage system to the second storage system in volume units.

3. A method of migrating data from a first storage system that is connected to one port of a switch (102) provided with a plurality of ports each assigned its own port identifier by means of a communication channel and receives, via the switch (102), read/write requests from a host computer (101) connected to another port of said switch (102) by means of a communication channel, to a second storage, the method comprising the steps of:

connecting the second storage system, by means of a communication channel, to a port, of said switch (102), other than the ports connected to the first disk unit and to the host computer (101);  
mutually exchanging the port identifier assigned to the port connected to the first storage system and the port identifier assigned to the

port connected to the second storage system; writing the data in the first storage system to the second storage system via the switch; if, from said host computer (101), there is a read or write request for the first storage system, sending the request to the second storage system; and,  
if the data for the read or write request has been written already to the second storage system, performing read or write processing of the already written data by means of the second storage system, and if the data for the read or write request has not yet been written to the second storage system, writing the data for the read or write request to the second storage system, and performing read or write processing of the written data by means of the second storage system.

4. A data migration method according to claim 3, further comprising the steps of:

directly connecting the first storage system and the second storage system by means of a communication channel; and,  
without passing through the switch, writing data in the first storage system to the second storage system via the communication channel that directly connects the first storage system and the second storage system.

5. A data migration method of migrating data from a first storage system that is connected to one port of a switch provided with a plurality of ports each assigned its own port identifier by means of a communication channel and receives, via the switch, read/write requests from a host computer connected to another port of said switch by means of a communication channel, to a second storage, the method comprising the steps of:

connecting the second storage system to a port, of the switch, other than the ports connected to the first storage system and to the host computer, by means of a communication channel;  
mutually exchanging the port identifier assigned to the port connected to the first storage system and the port identifier assigned to the port connected to the second storage system;  
reading information on logical volumes constructed in the first storage system;  
constructing the same number and same sizes of logical volumes as logical volumes that have been constructed in the first storage system in the second storage subsystem;  
writing data stored in the first storage system to the second storage system via the switch;

managing completion of said writing from the first storage system to the second storage system in volume units;

if, from said host computer, there is a read or write request for the first storage system, sending the request to the second storage system; and,

if the data for the read or write request has been written already to the second storage system, performing read or write processing of the already written data by means of the second storage system, and if the data for the read or write request has not yet been written to the second storage system, writing the data for the read or write request to the second storage system, and performing read or write processing of the written data by means of the second storage system.

6. A data migration method according to claim 5,

wherein the second storage system reads, via the switch, the logical volume information from the first storage system and manages the completion of said writing to the second storage system in volume units.

7. A data migration method according to claim 5,

wherein the switch reads the logical volume information from the first storage system, constructs the same number and same sizes of logical volumes as logical volumes that have been constructed in the first storage system, in the second storage system, and manages the completion of said writing to the second storage system in volume units.

8. A data migration method according to claim 5,

wherein an information processing unit connected to the switch reads the logical volume information from the first storage system, constructs the same number and same sizes of logical volumes as logical volumes that have been constructed in the first storage system, in the second storage system, and manages the completion of said writing to the second storage system in volume units.

9. A data migration method of migrating data from a first storage system that is connected to one port of a switch provided with a plurality of ports each assigned its own port identifier by means of a communication channel and receives, via the switch, read/write requests from a host computer connected to another port of said switch by means of a communication channel, to a second storage, the method comprising the steps of:

connecting the second storage system to a port, of the switch, other than the ports connect-

ed to the first disk unit and to the host computer, by means of a communication channel; switching the destination of a read/write request from the host computer from the first storage system to the second storage system; writing data in the first storage system to the second storage system via the switch; and, if the data for a read or write request from the host computer has been written already to the second storage system, performing read or write processing of the already written data by means of the second storage system, and if the data for the read or write request has not yet been written to the second storage system, writing the data for the read or write request to the second storage system, and performing read or write processing of the written data by means of said second storage system.

10. A data migration method according to claim 9, further comprising the steps of:

reading the number of logical volumes constructed in the first storage system and their sizes;

constructing the same number and same sizes of logical volumes as logical volumes that have been constructed in the first storage system in the second storage system; and, writing, in volume units, data stored in the first storage system to the second storage system via the switch.

11. A data migration method according to claim 10, wherein the second storage system manages the completion of said writing to the second storage system in volume units.

12. A data migration method according to claim 1, wherein the communication channel connecting between the switch and the first storage system, the communication channel connecting between the switch and the second storage system and communication channel connecting between the switch and the host computer are fiber channels.

13. A disk storage system to which data stored in other storage system connected to a switch is migrated via the switch, comprising:

a port for connecting to the switch, a program for reading data stored in the other storage system via the switch, if data for a read or write request from a host computer that is connected to the switch has been read already, performing read or write processing of the already read data, and if the data for the read or write request has not yet been read, reading the

data for the read or write request from the other storage system.

14. A computer connected to a switch to which a first storage system is connected, comprising:

5

a port for connecting to the switch,  
a program, in case that a second storage system to which data stored in the first storage system is migrated via the switch is connected to the switch, for changing the destination of a read or write request from the first storage system to the second storage system.

10

15

20

25

30

35

40

45

50

55



Fig 1

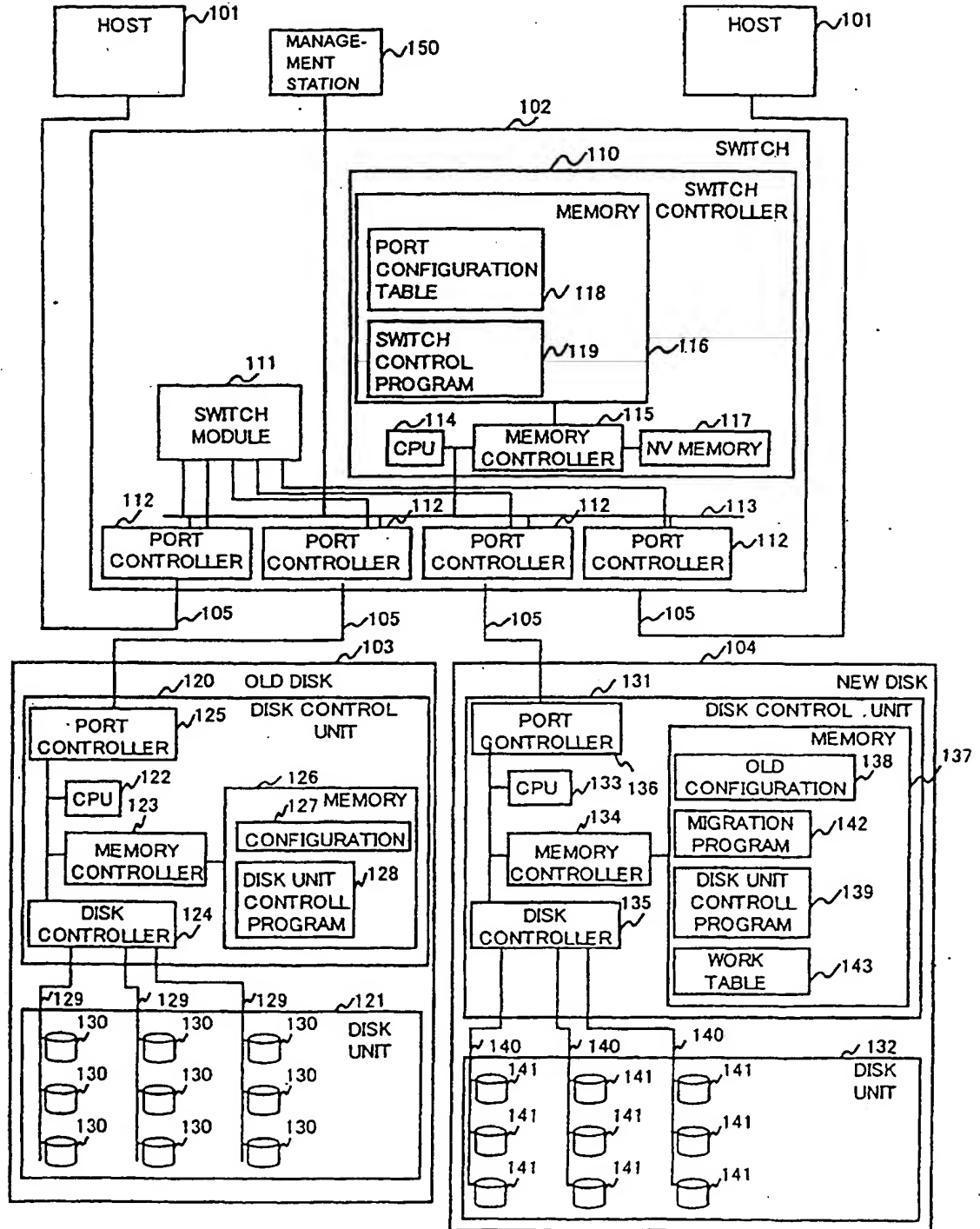


Fig 2

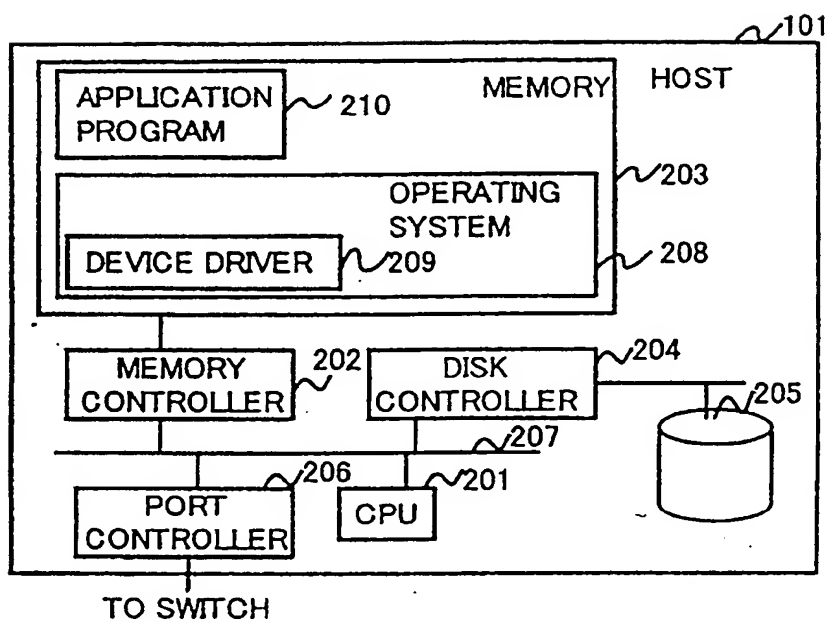


Fig 3

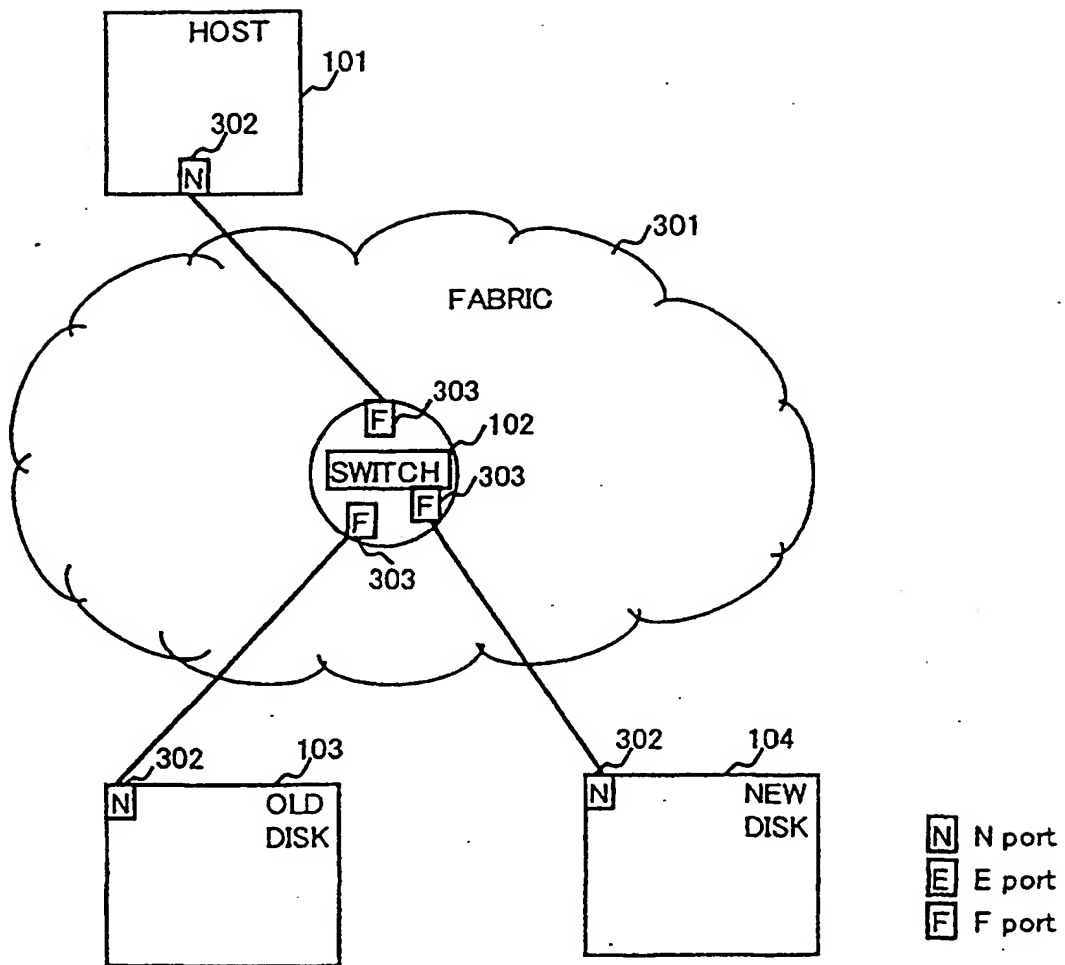


Fig 4

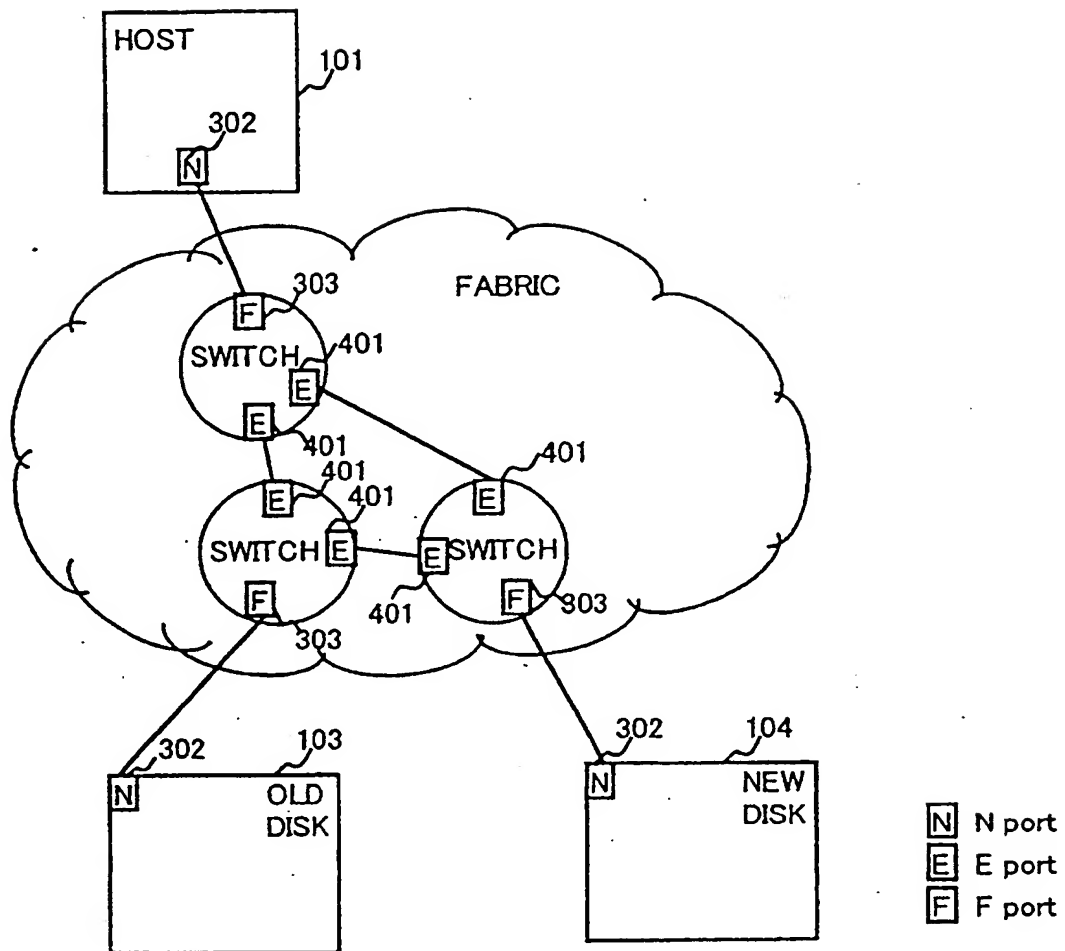


Fig 5

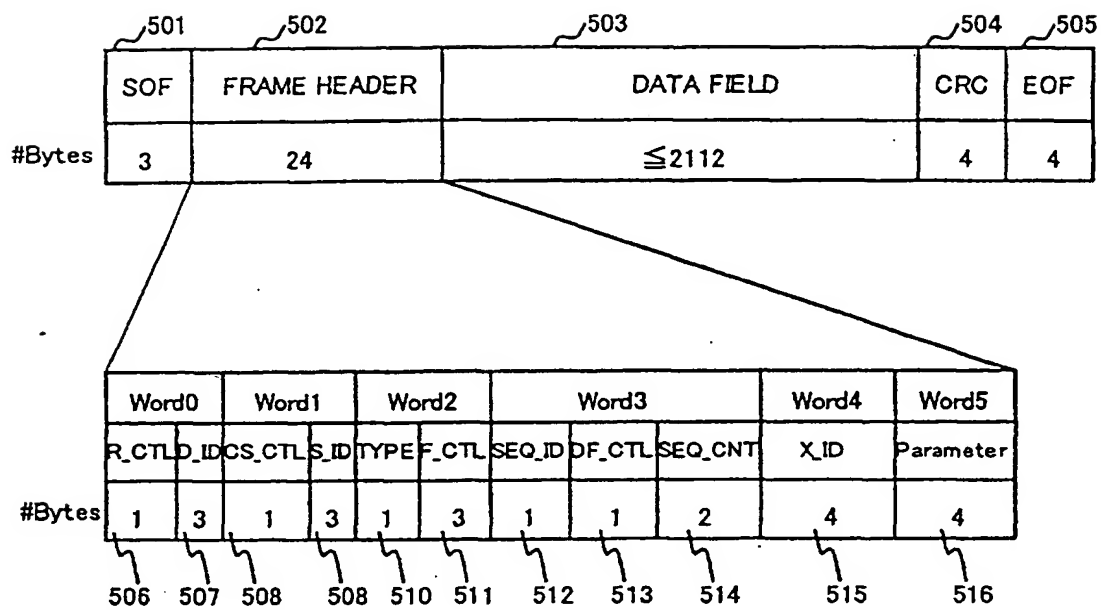


Fig 6

601

602

Logical Port ID	Physical Port ID
0	0
1	1
2	2
3	3
⋮	⋮
N-1	N-1

(A) Before Port Switching

601

602

Logical Port ID	Physical Port ID
0	0
1	2
2	1
3	3
⋮	⋮
N-1	N-1

(B) After Port Switching

Fig 7

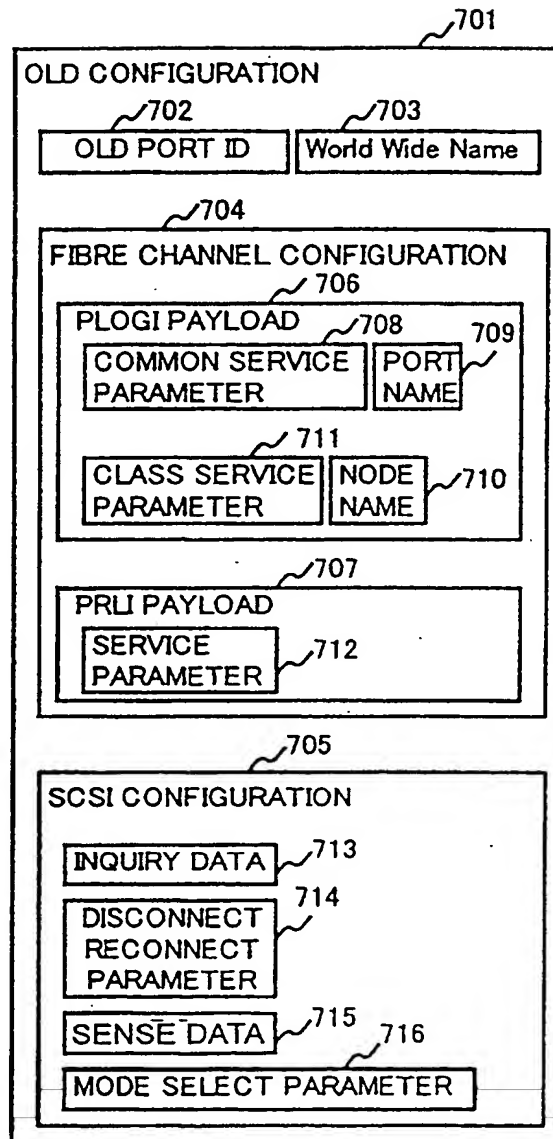


Fig 8

801 VOLUME NUMBER	802 SLOT NUMBER	803 STATUS
1	10	MIGRATING
1	25	DONE
1	26	DONE
2	15	DONE
2	16	DONE
⋮	⋮	⋮



Fig 9

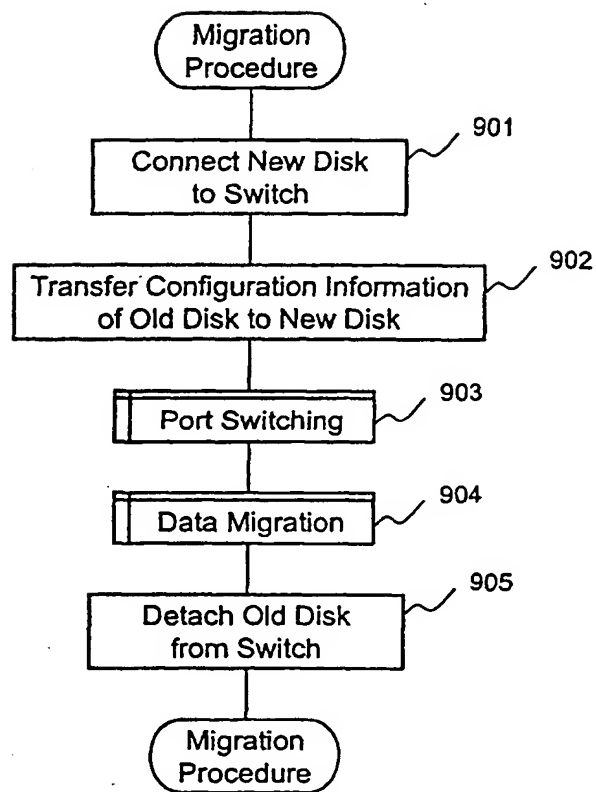


Fig 10

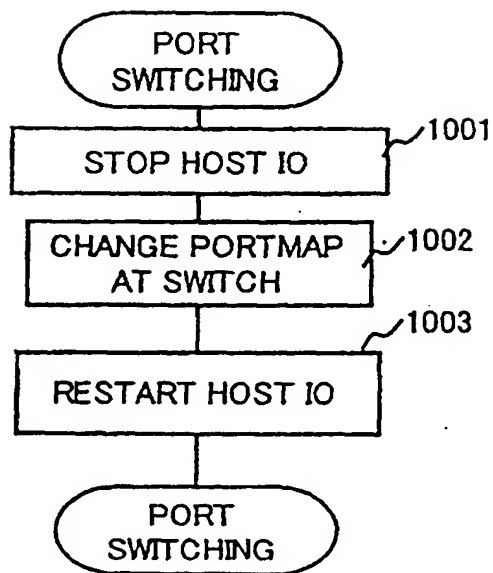


Fig 11

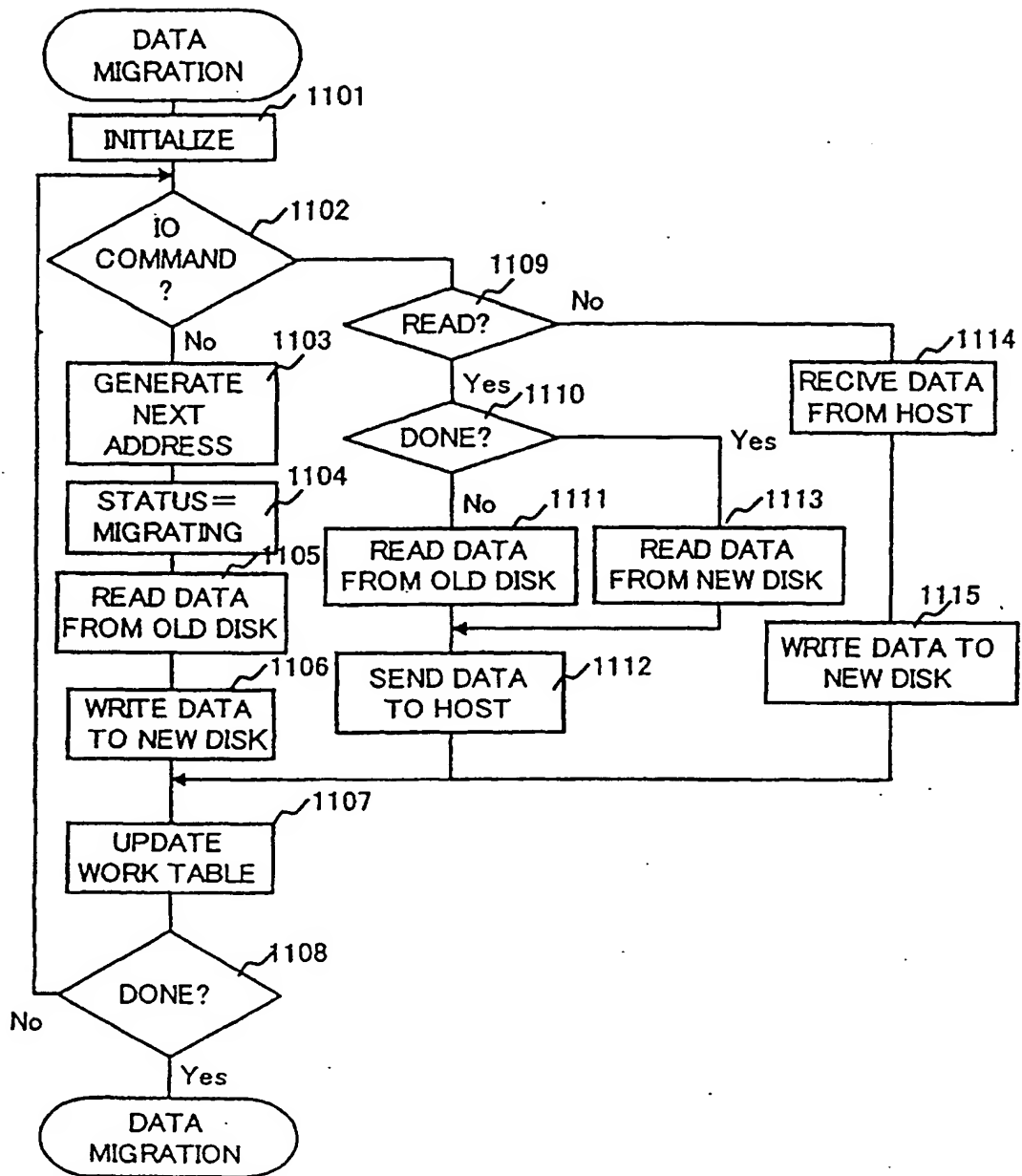


Fig 12

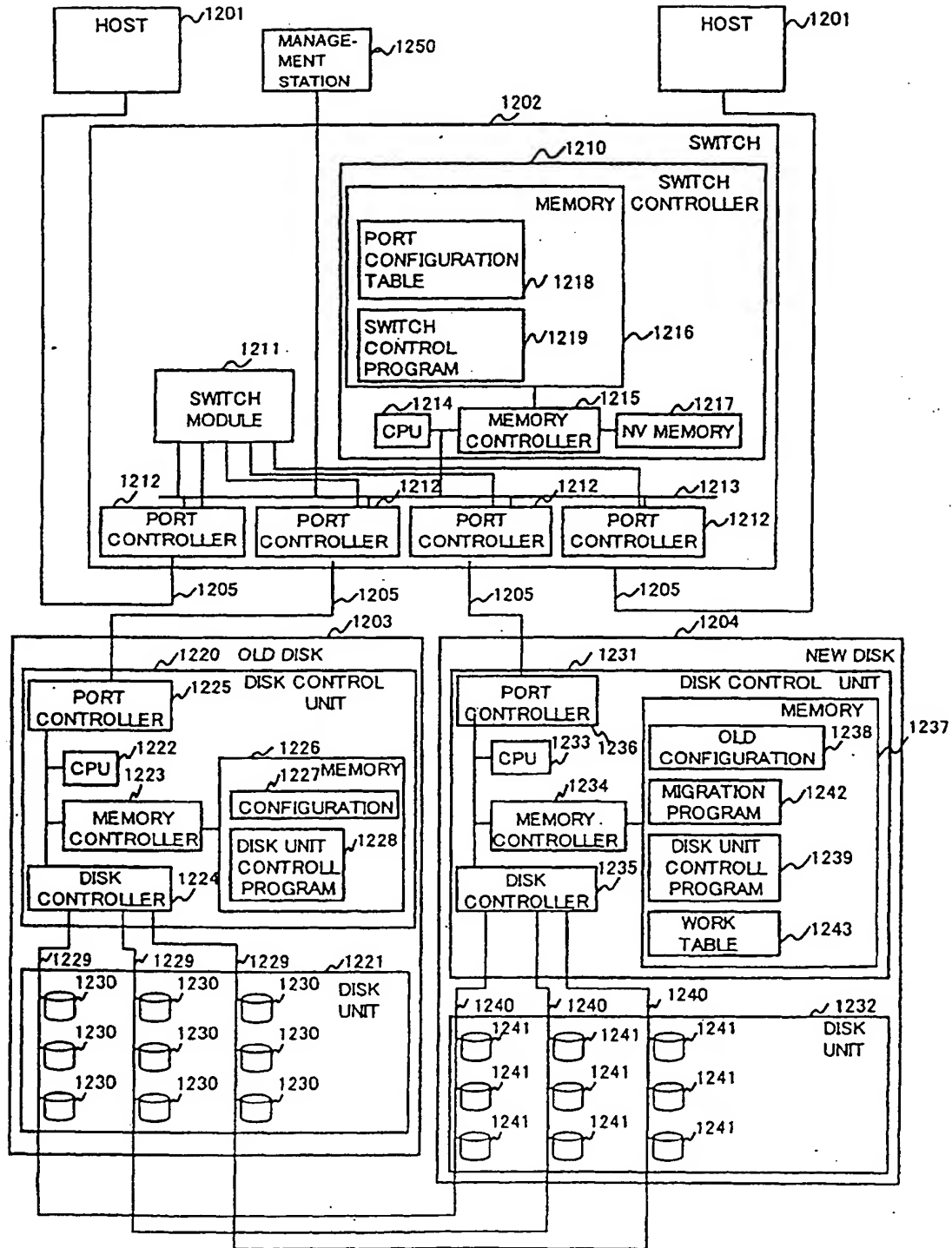


Fig 13

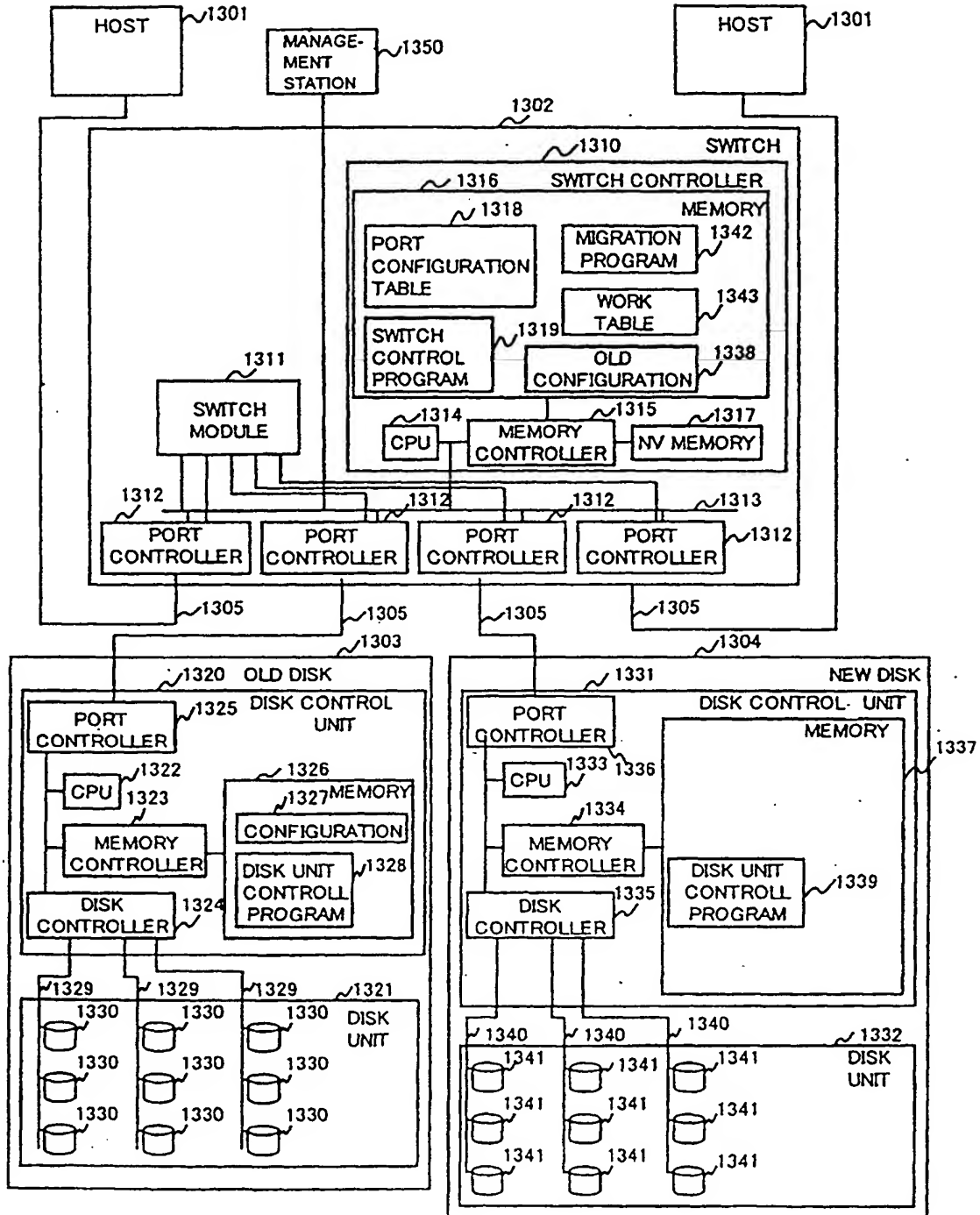


Fig 14

Logical Port ID	Physical Port ID	Status
0	0	Normal
1	2	Migrating
2	1	Normal
3	3	Normal
⋮	⋮	⋮
N-1	N-1	Normal

Fig 15

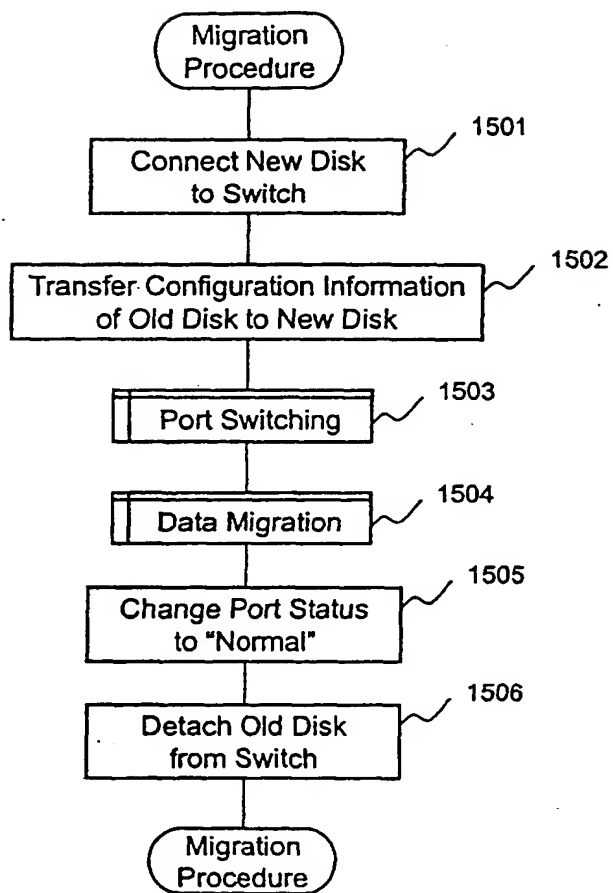


Fig 16

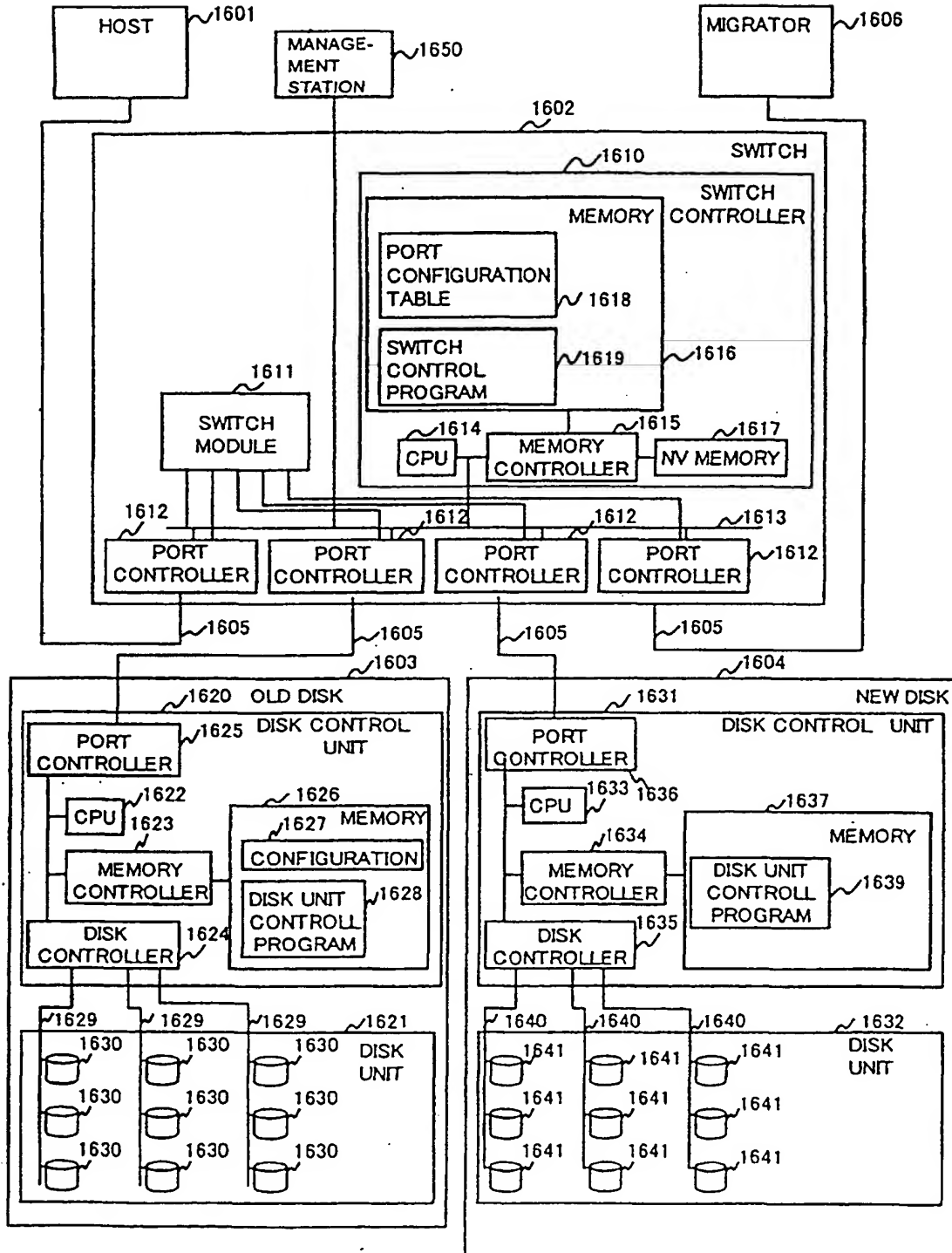




Fig 17

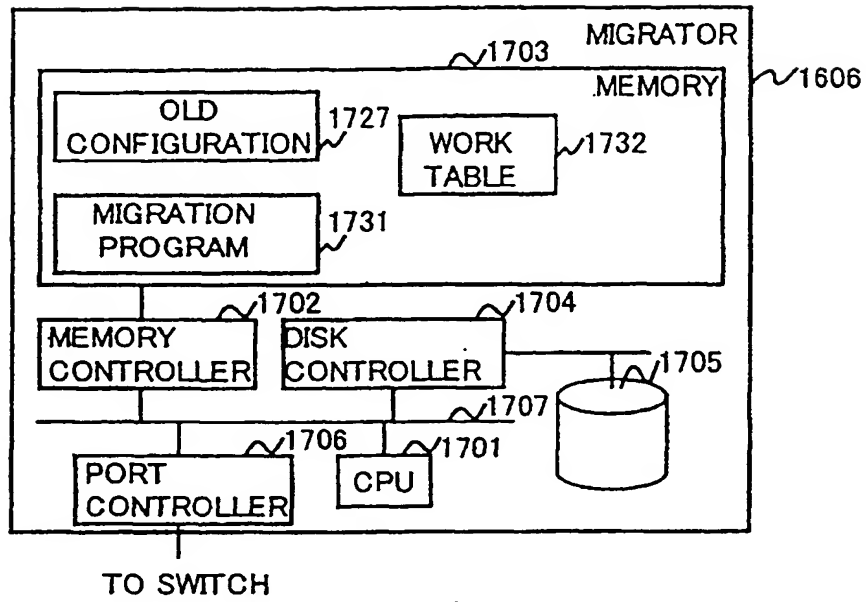


Fig 18

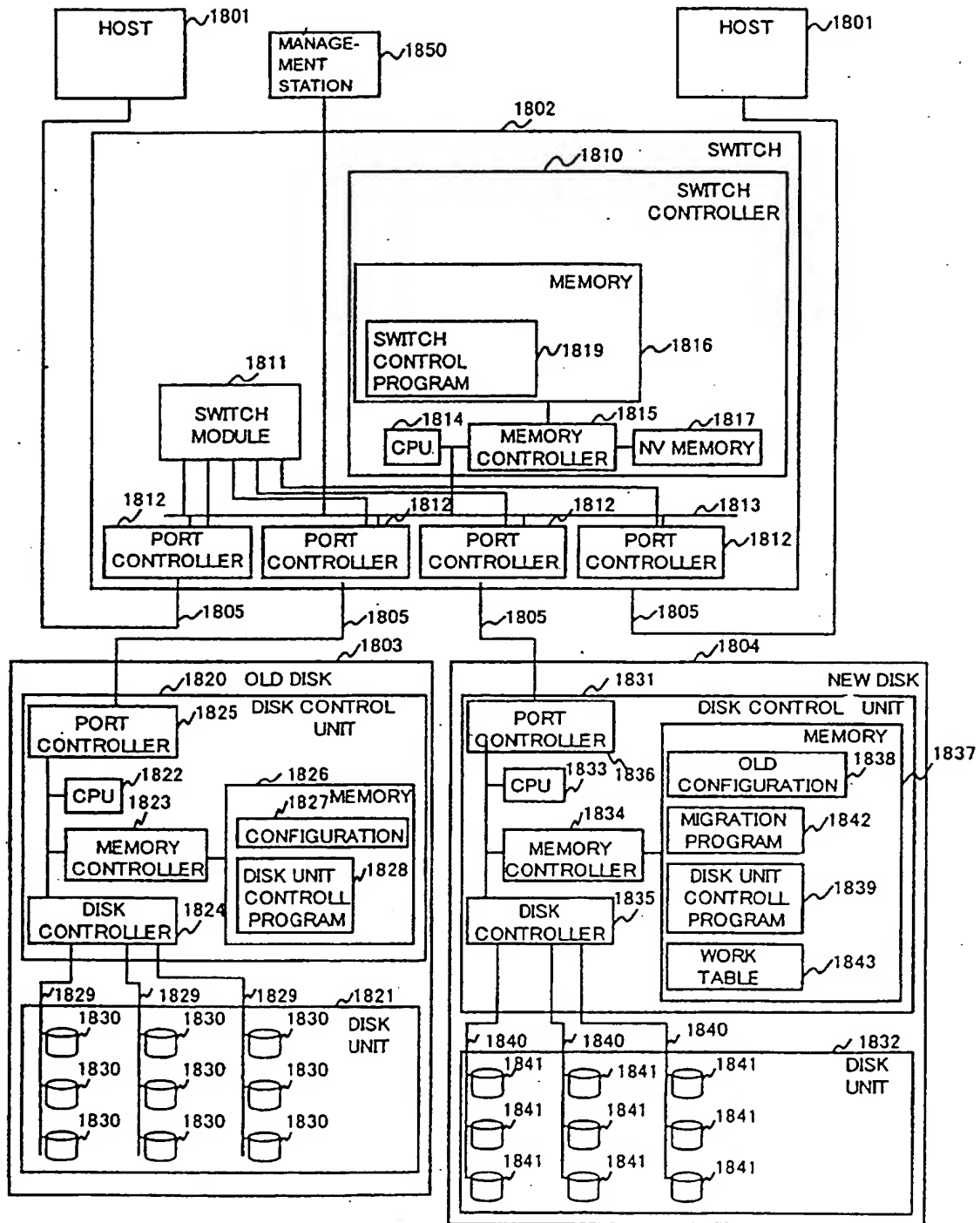


Fig 19

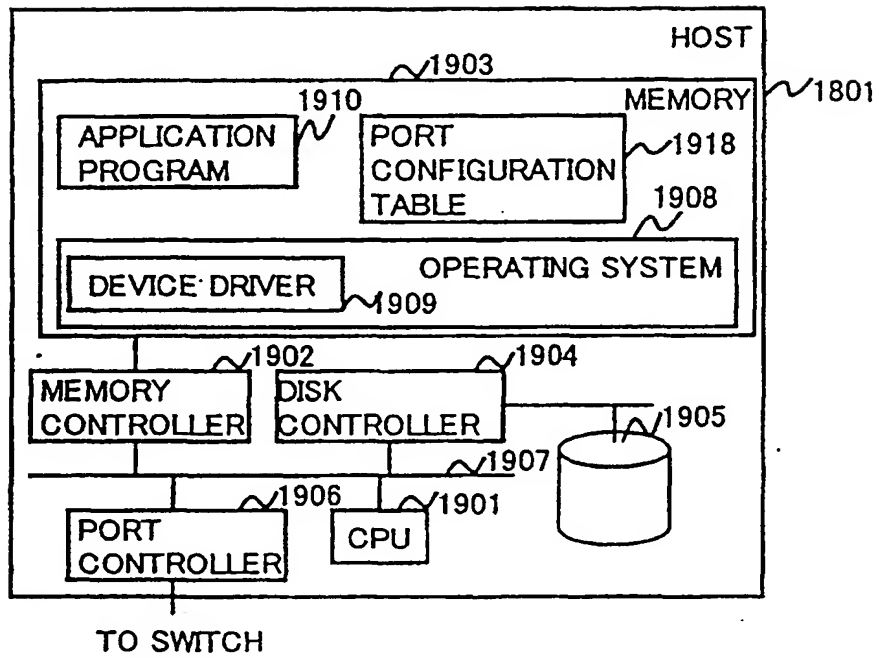


Fig 20

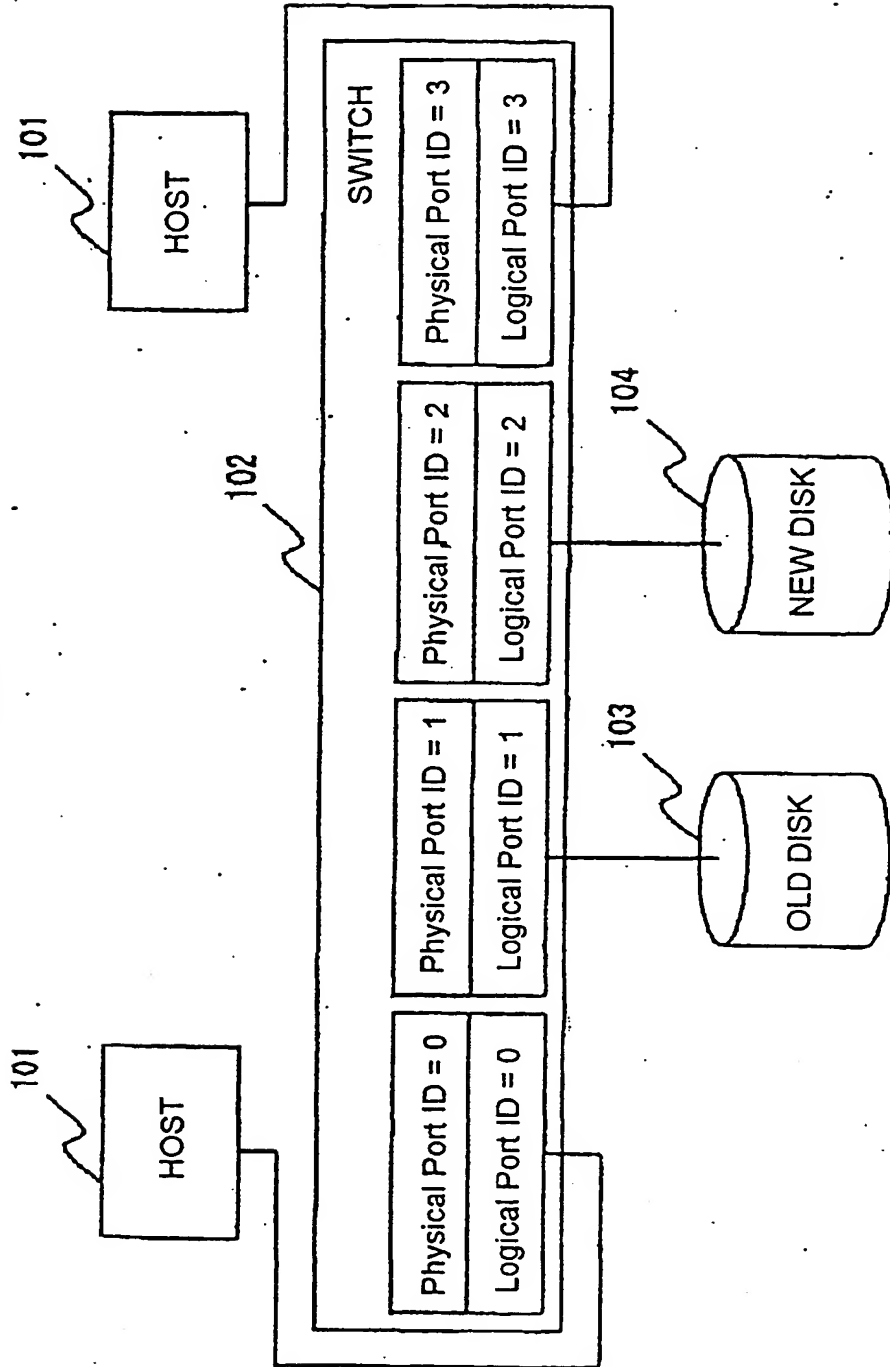


Fig 21

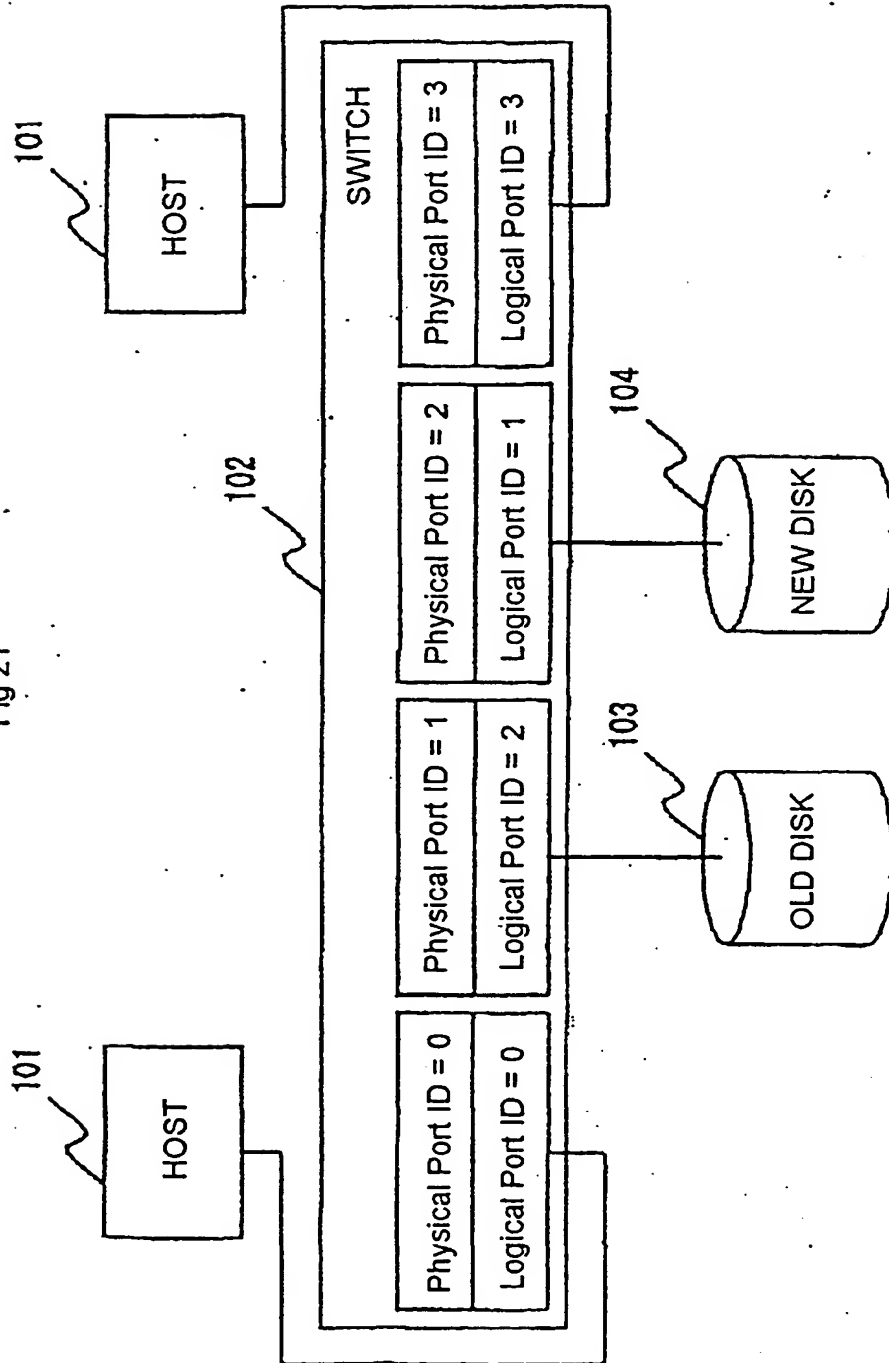


Fig 22

